ZX COMPUTING MONTHLY

AN ARGUS SPECIALIST PUBLICATION JUNE 1987 £1.50

FOR ALL SINCLAIR USERS



They Came From Within!

BUGS IN THE ROM

The true story of the Spectrum's built-in faults

MULTI-SCROLL UTILITY

JOHN RITMAN INTERVIEWED

128 MACHINE CODE

MONSTER HITS:

HEAD OVER HEELS

ENDURO RACER

WIN!

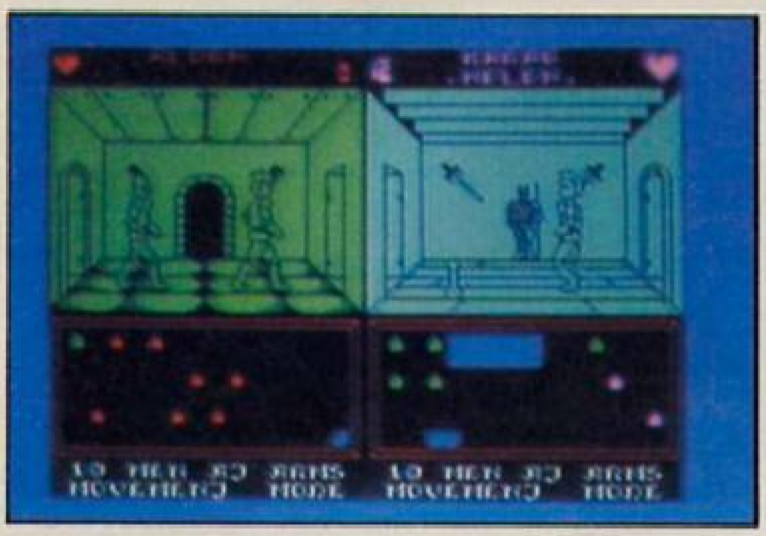
HEWSON'S GUNRUNNER

CODEMASTERS GAME PACK


"Addictive and compelling — you just can't put your joystick down."
TAITO COIN-OP
TAITO COIN-OP
SLAP FIGHT
TM
From Taito Corporation comes the last word in planet combat . . .
Multi screen, realistic graphics and all the coin-op features.
SPECTRUM COMMODORE
£7.95 £8.95
SPECTRUM AMSTRAD
Imagine
..the name of the game
Arcade GOLD
Imagine Software Ltd., 6 Central Street, Manchester M2 5NS. Telephone: 061-832 6633. Telex: 669977
Licensed from © Taito Corp., 1986 Programmed for Amstrad, Spectrum, Commodore by Imagine Software.

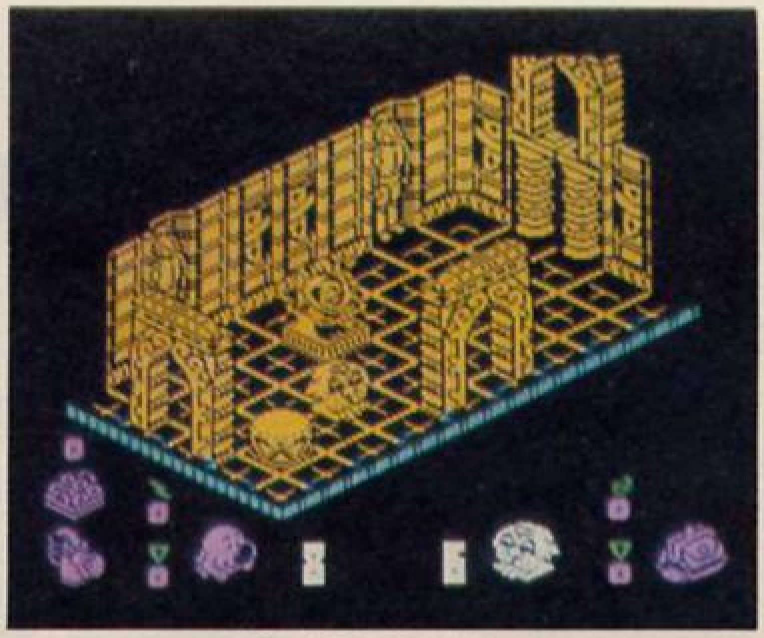
FEATURES

REVIEWS



Throne of Fire (41)

COMPETITIONS AND OFFERS



Head Over Heels (50)

CONTENTS

REGULARS

PROGRAMMING PROJECTS

NEWS



Catch 23

That's the title of the next release from Martech — a vector graphics blastathon in which you have to penetrate a heavily guarded military complex and steal the plans for a deadly satellite. As well as the combat aspect there is also an adventure element in the game as there are numerous locations to explore, objects to collect and puzzles to solve, so it should be more than just another vector graphics shoot 'em up. Catch 23 due out about now, priced at £8.95.

Marble Madness Winners

There are twenty five winners in the Marble Madness Competition. They each receive a copy of Melbournes Houses Marble Madness Construction set.

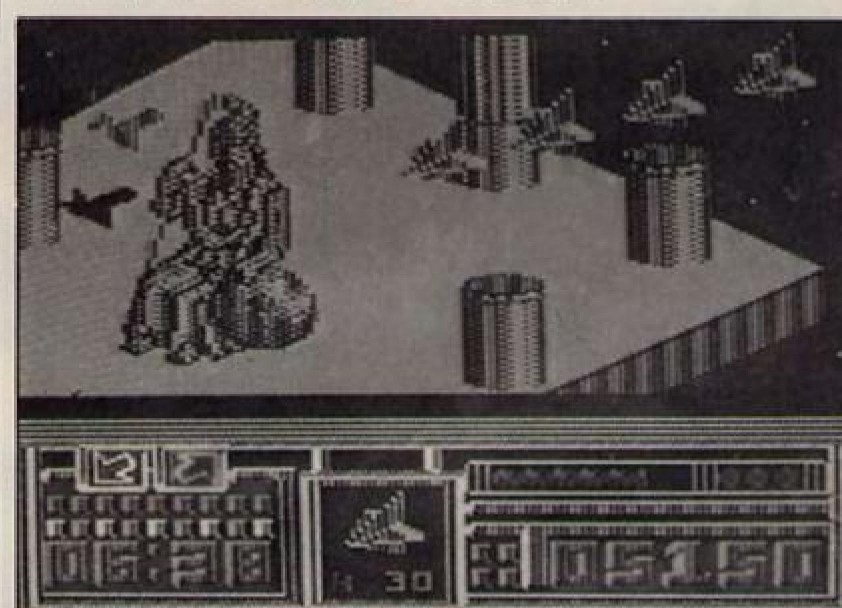
They are: B. Hobson, Huddersfield; D. Pike, Saltcoats; G. Dunnet, Ontario, Canada; T. Denks, Amsterdam; M. Richardson, Aylesbury; C. Welsh, Glasgow; R. Hand, Ilford; G. Galloway, Liverpool; A. Tindall, Pickering; S. Vriezers, Haarlem, Netherlands; M. Kemp, Westerham; L. Chorlton, Bude; D. Cole, Haywards Heath; C. Haggart, Macduff; S. Elsinga; Harlingen, The Netherlands; D. Luby, Duns; H. Neison, Edinburgh; P. Francis, Hantes-Wiheries, Belgium; R. Grant, Boness; R. Patrick, Gent, Belgium; T. French, New Milton. P. Evans, Chichester; P. Barker, Gillingham; P. Wright, Benfleet; B. Perry, Heightington.

Cosmic Shock Absorber Winners



The thirty winners of our Cosmic Shock Absorber Competition are:

B. Coles, Milton Keynes; D. Jenks, Sherburn-in-Elmet; K. McGrath, Foulkhills; K. Rosser, Barry; L. Martin, Basingstoke; D. Greenwood, Bury; A. Cousins, Peterborough; B. Jefferson, Scarborough; S. Booth, Leeds; H. McGregor, Frome; C. Dyball, Gt Yarmouth; P. Neal, Reading; J. Williams, London SW8; S. Mennet, Skelmersdale; U. Yasin, Leeds; P. McNicol, Glenrothes; N. Horsfield, Barneswick; T. Smith, Leicester; M. Palmer, Bristol; B. Rumbelow, Deal; S. Grimley, Ashford; S. Postgate, Guisborough; L. Millea, Liverpool; A. Fletcher, Derby; K. Dryburgh, Kircaldy; A. Dalki, London SW8; V. Lawlor, Wexford; L. Halls, Derby; S. Jeffery, Edgware; I. McVicar, Clydebank.



Leviathan

English Software, who normally move in the more rarified field of Atari and Commodore software, have converted their game, Leviathan, for the Spectrum. Modestly described as "the shoot 'em up to end all shoot 'em ups", Leviathan lets you pilot your spaceship over a 3D landscape as you battle the usual horde of aliens. The 128 version will have an enhanced soundtrack, and I'm told that the CBM 64 version was good, but very hard. See for yourself this month if you've got £7.95 to spare.



T.S.A.M. at last!

Taking over the 'better late than never' throne (recently vacated by Ocean) Ariolasoft have finally gotten their Christmas '86 release, They Stole A Million ready to go into the shops. Or so they tell us, after all they said that before Christmas. Still, while you're all waiting eagerly for the game's arrival here's a picture of Ariolasoft's Mark Eastman inprisoned in solitary confinement until the game is released.

Icon Graphix Winners

Twenty five ZX readers carry off copies of Audiogenics Icon Graphix utility. They are:

A. Lewis, Newtown, Powys; I. Laing, Richmond, N. Yorks; P. Clark, Poole; S. Oliver, Macclesfield; S. Wing, London, SE23; E. Dron, Sutton; R. Davey, Bournville; S. Troung, Mansfield; W. Johnston; Leicester; B. Matskevich, Hod Hasharon, Isreal; J. Parker, Plymouth; J. Norton, Ashinton; R. Sails, Sale; A. Hogg, BFPO17; D. Shepherd, Bristol; C. Homow, Hove; R. King, Saltyash; D. Orusum, Stoke; S. Hollinshead, St Ives; V. Watson, Darwen; A. Inglis, Carlisle; A. Cook, Copthorne; M. Teeger, London NW6; J. Johnson, Burfod; P. Atkins, Lochgilphead.

Errata

Egg on face time I'm afraid. somewhere along the line a couple of numbers vanished from the listing in our Memory Expansion Program in the April issue.

The final number on line 230 should have been 561, the final number in line 250 should have been 609. In line 270, the number 1791051 should have read 179,1051 and in line 290 the final number was 790.

Apt perhaps that our April Fool program should backfire on us.

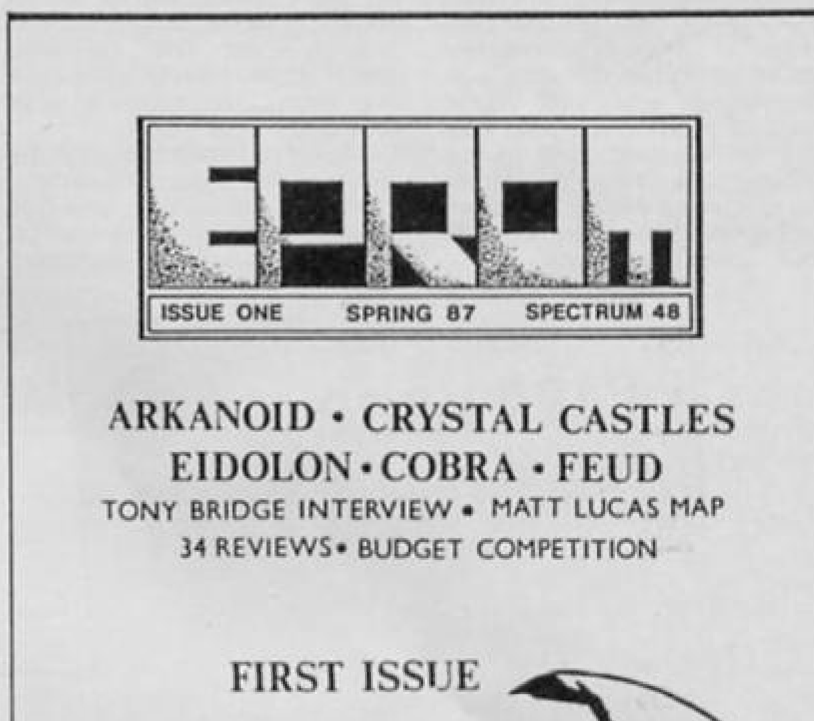


Rogue Trooper Winners

The winners in our Rogue Trooper competition who created vile, evil adversaries for the 2000AD comic hero to annihilate are:

S. Clark, Retford; A. Dalli, London SW8; A. Wood, Harare, Zimbabwe; M. Doyle, Co Wexford; S. Drearley, Halifax; D. Fraser, Mallaig; G. Browse, Burwell; S. McDonald, Hovik, Norway; B. Neal, Swansea; G. Proctor, Craigavon; M. Scott, Gateshead; A. Deacon, Amersham; M. Smith, Torpoint; C. Davies, Tupsley; C. Welsh, Glasgow; K. Ilyas, Manchester; D. Patrick, Geraardsoergen, Belgium; J. Rattenbury, Malmesbury; M. Brindley, Derby; P. Serbert, Harrowgate; S. Burnett, Enfield; T. Moore, Rosbrien; S. Smith, Transvaal; S. Turner, Isle of Wight; R. Grant, Boness; K. Young, Banbury.

They each win a copy of Piranha's game.



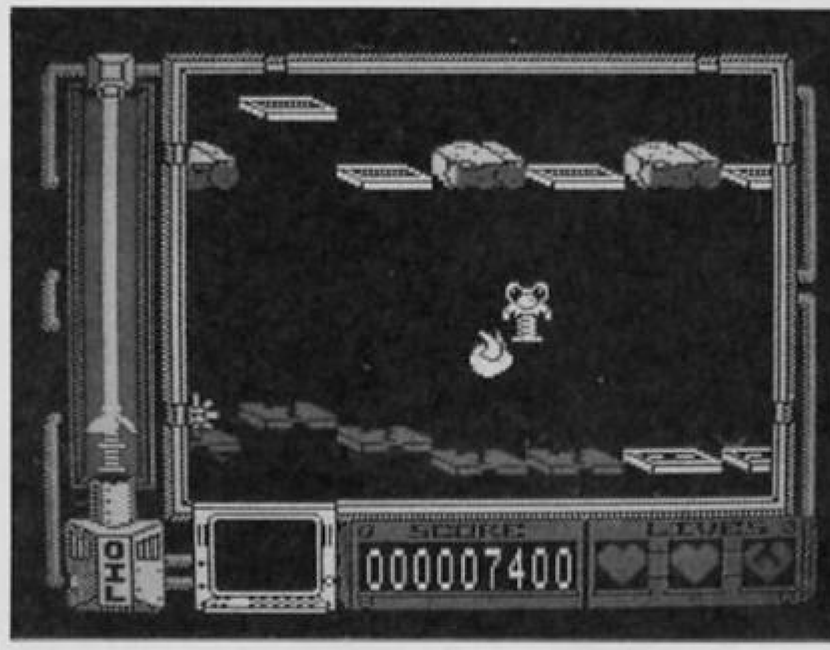
Eprom

We've just received a copy of EPROM a new software fanzine produced by Tony Worrall. It's mostly games reviews, along with a few more general pieces, charts, adventure section an so on, and is produced totally by Tony from his home in Preston and costs just 80p. It's quite well done, and any magazine that gives Arkanoid a rating of 90% can't be bad. EPROM is available from Tony Worral, 328 The Maltings, Penwortham, Preston, Lancs PR1 9FD.

Apocalypse when?

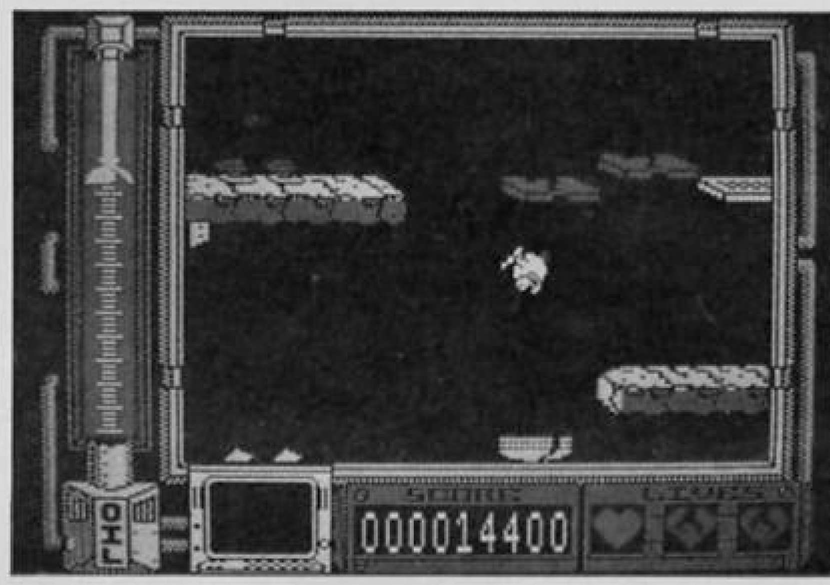
Apocalypse, the best wargame ever produced for the Spectrum is about to make a comeback. The game has had a bit of a chequered past, originally programmed by Red Shift, marketed by someone or other from their back bedroom, then bought out by someone else, it's now due to reappear on the Command Software label, which is a subsidiary of french outfit Infogrames.

The game involves upto four players who can stab each other in the back as they plot to take over a variety of different scenarios. The last resort is, of course, a nuclear strike, at which point the computer makes all sorts of fun noises and flashing effects on the screen. The new, upgraded, version of Apocalypse is priced at £9.95 and is due out this month. If I remember rightly the game is even endorsed by CND, so it's ideologically sound too.



The Thing behind the Mask

Gremlin's next two games are Thing Bounces Back and MASK, based on the comic strip of the same name. Thing is actually bouncing on to the Spectrum for the first time as the original Thing game wasn't available in that format. The game is a platform game of sorts in which the Thing bounces around on a spring in an attempt to halt an overproductive toy factory. Meanwhile, Mask takes you on a hunt through time and space to search for your companions who have been scattered by the evil organisation VENOM. Both games should be out around May/June.



Thrills and spills!

The big news this month is of the imminent arrival, hot from the arcades, of two of the fastest, zappiest games your Spectrum is ever likely to have coursing though its microcircuitry. Pass Your Driving Test from Audiogenic is a truly awesome tutorial program dealing with the sorts of questions you're likely to face when you take your driving test. Thrill to the kerb-bumping three point turn, the gear-crunching hill start. And then, when you thought your sizzling circuitry could stand no more you can load up the latest state-of-the-art program, General Election from Lobos Software. Designed for those election night parties, this program (designed for up to 350 players) lets you guess the outcome of the general election and then tells you and the other 349 players who made the best guess.

Apart from the fact that the result of the election seems a foregone conclusion, you can contact Lobos Software on 0274-498804 if you're interested.



Micro-Source

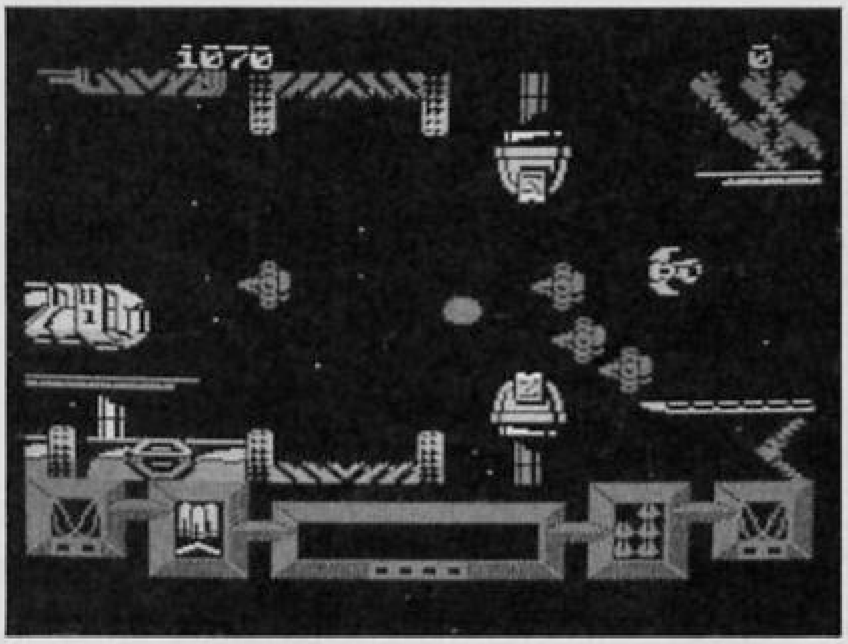
Last year we reviewed a small hardware add-on called the Micro Source. Originally produced by Currah, this combined FORTH/Assembler development tool was taken over by Quad House but supplies apparently never made it into the shops. The Micro Source has now been taken up again, this time by Simmons Electronics (0970-4866). Aimed at the more serious programmer, it got a good review the last time around and priced at £19.95 is good value. Contact Simmons Electronics for more information.

Murder on the Atlantic

Infogrames are heading into Agatha Christie territory with their next adventure release. Murder on the Atlantic puts you on an ocean cruiser in the 1930's where a murder has just been committed. Donning your Hercule Poirot moustache you have to solve the whodunnit mystery in a manner similar to Infogrames' previous adventures, Vera Cruz and The Sydney Affair. Murder on the Atlantic will cost £12.95 (false moustache costs extra).

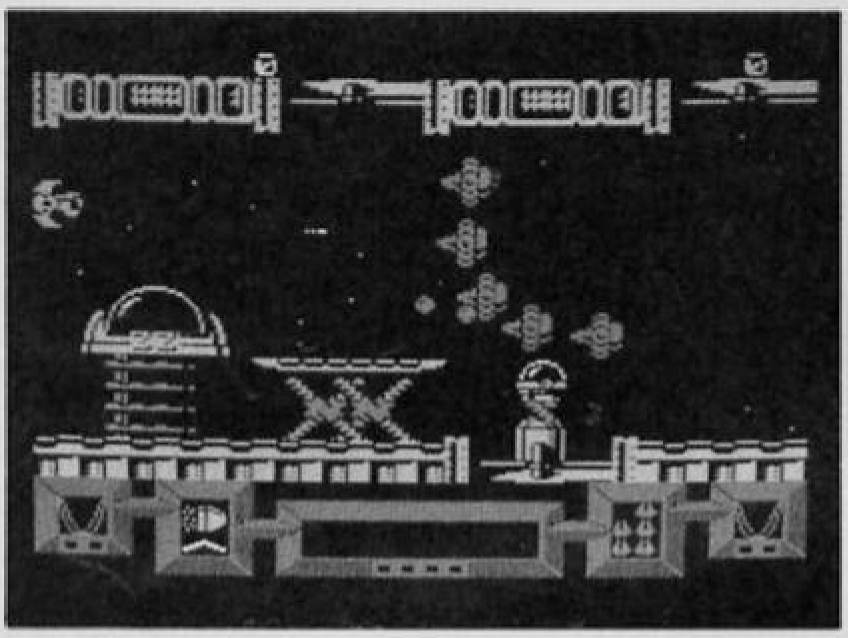
Big Bang

PSS are releasing a compilation tape containing three of the titles in their Wargamers series. The compilation, called Conflicts 1, will include Battle of Britain, Theatre Europe, and Falklands '82 and is due for a May release, priced at £12.95.



Hewson strip

Hewson are steaming away at a fair old rate these days, and hardly an issue goes by without a new Hewson game to review. Next on their Spectrum schedule is Zynaps, a spacey shoot 'em up presented in a comic strip format "to give it that all-important feeling of continuity." Written by Dominic Robinson and John Cumming, Zynaps should be out in June for £7.95.

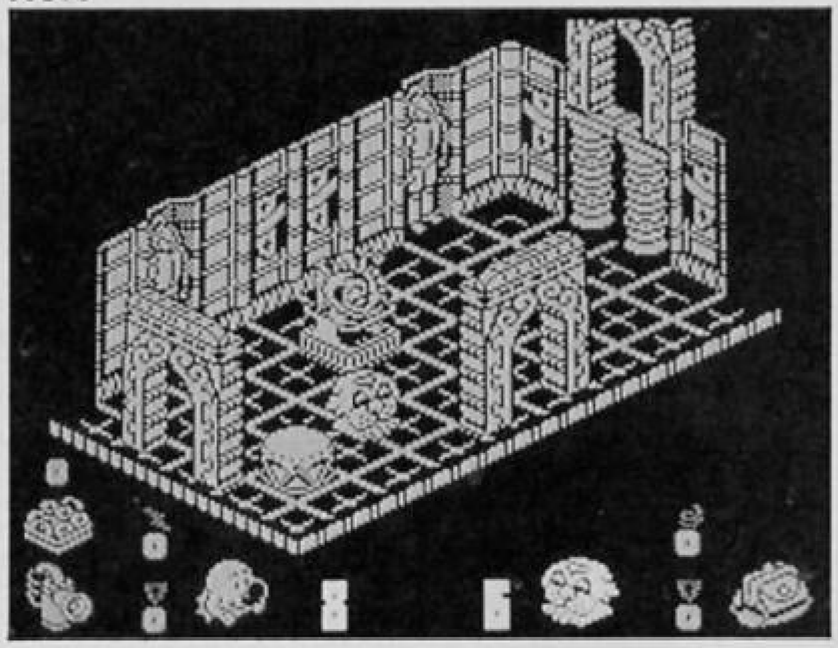




Whitehouse cometh

Mary Whitehouse, that is. Because you can bet your boots that as soon as word gets out about the decapitation scene in Palace Software's new Barbarian game the 'ban the video nasties' brigade are going to be up in arms. But Palace remain unrepentant and insist that not only is Barbarian the best swordfight game that yet been produced, it's also just what the kids in the arcades will want to play. And they're probably right — it's just a good thing that they don't intend to use the nude Maria Whittaker sprite after all...

No.5

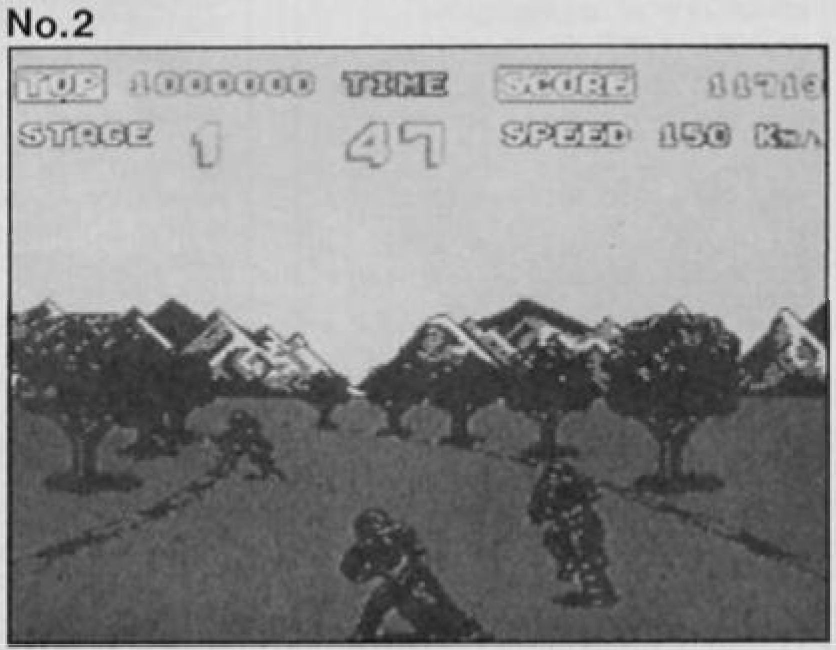


Spectrum Games Top Ten

1	( )	Six Pak Hit Pack	Elite
2	( )	Enduro Racer	Activision
3	(3)	Gauntlet	US Gold
4	( )	Super Soccer	Imagine
5	( )	Head Over Heels	Ocean
6	( )	Into The Eagle's Nest	Pandora
7	(6)	Paperboy	Elite
8	(5)	Konami's Coin Op Hits	Imagine
9	( )	Footballer of the Year	Gremlin
10	( )	Four Great Games	Tynesoft

(Chart supplied by W. H. Smith)

No.2



Cop Out Winners

The ten first prize winners in our Cop Out competition who win a copy of the Mikrogen game plus a calculator are:

B. Shimmings, St Leonards-on-Sea; A. Elsegood, Grantham; I. Drury, Diss; N. Owen, High Wycombe; S. Lyndsell, Ramsgate; J. Bell, Wakefield; M. Roddy, Leeds; S. Davie, Glastonbury; S. Clark, Retford; K. Young, Banbury.

Ten runners-up win copies of the game. They are: L. Ryan; I. Logan, Paisley; P. Gibson, Sunderland; C. White, London, E16; A. Brown, Northampton; J. Smith, Barnsley; T. Marshall, Kent; P. Luckham, Manchester; C. Jones, Guildford; S. G. Fernandes, Rio De Janeiro, Brazil.

SHORTCUTS

This month's selection of readers' mini-routines.

M/C break 2

```
  1 REM M/C break Mk2
 10 CLEAR 65278: RESTORE : FOR
a=65279 TO 65322: READ b: POKE a
,b: NEXT a
 20 DATA 1,255,245,205,84,31,48
,4,241,195,56,0,241,62,2,205,1,2
2,1,6,24,205,217,13,193,205,43,4
5,205,227,45,237,86,255,195,3,19
,62,254,237,71,237,94,201
```

Was sent in by Ray Reeves and is an amendment of an earlier program by A. Vellacott.

This program is useful when programming in machine code should you inadvertently set up an infinite loop. The keyboard is scanned by using the interrupt and if BREAK is pressed then it returns to BASIC. So far this is the same as the program we used previously, but what Ray has done is add some code so that the address at which the program was broken is printed in the top right hand corner. This is much more useful as you now have some idea of where to look in your program to debut the problem loop.

Horizontal/vertical flip

Two screen routines from Paul Extance of Lincoln which may liven up your own program.

They both do exactly as they say, the screen is "flipped" either horizontally or vertically depending on which routine you use. The built-in demo is an effective way of seeing the routines in action.

```
 10 REM Horezontal flip
 20 FOR f=50000 TO 1e9: READ a:
 IF a<>999 THEN  POKE f.a: NEXT
f
 30 DATA 33,0,64.17.0.200.1.0,2
7,237,176
 40 DATA 33,0,200,78,6,8,203,17
,31,16,251.119,35,124.254,224,32
,241
 50 DATA 33,255,90,17,0,224,6,6
,197,6,128,26,119,43,19,16,-6,19
3,16,-12
 60 DATA 33,255,87,17,0,200,6,4
8,197,6,128,26,119,43,19,16,-6,1
93,16,-12,201,999
 100 REM Vertical flip
 110 FOR f=50100 TO 1e9: READ a:
 IF a<>999 THEN  POKE f,a: NEXT
f
 120 DATA 33,0,64,17,0,200,1,0,2
7,237,176
 130 DATA 33,0,200,78,6,8,203.17
,31,16,251,119,35,124,254,224.32
,241
 140 DATA 33,0,200,6,216,197,84,
93,6,32,126,245,35,16,-5,98,107,
6,32,241,119,35,16,-5,193,16,-22
 150 DATA 17,0,64,33,0,200,1,0,2
7,237,176,201,999
 200 PRINT AT 10,10;"DEMO": FOR
i=1 TO 100: PRINT AT RND*21,RND*
30; INK RND*7;"X": NEXT i
 220 PRINT #0;AT 1,0; INVERSE 1;
"Horizontal flip": RANDOMIZE USR
 50000
 230 PAUSE 0
 240 PRINT #0;AT 0,0; INVERSE 1;
"Vertical flip": RANDOMIZE USR 5
0100
 250 PAUSE 0: GO TO 220
```

Break ignore

Is from regular R. Glavas. Type it in and initialise it by RANDOMIZE USR 50000. You can always save the code by SAVE "code" CODE 50000,42 and re-load it at the start of your own program.

Once the code has been initialised then pressing the BREAK key has no effect, normal and "official" program stops are allowed. This could be useful as protection for your games or as a trap against accidental pressing of the break key.

```
  1 REM Break ignore
 10 CLEAR 49999: RESTORE : FOR
a=50000 TO 50041: READ b: POKE a
,b: NEXT a
 20 DATA 59,59,225,17,15,0,25,2
35,42,61,92,115,35,114,201,59,59
,33,58,92,126,254,20,40,9,254,12
,40,5,51,51,195,3,19,54,255,35,2
03,254,195,125,27
 100 REM Example
 110 LET l=USR 50000
 120 PRINT "Try Break !"
 130 FOR f=1 TO 21: PRINT AT f,0
; INK f/4;f
 140 CIRCLE 128-f*2,88,f*3: NEXT
 f
 150 STOP
```

Inverter/wrap around

Another pair of effects for your screen, this time supplied by John Fryc of Mansfield.

Both are short and relocatable machine code routines which are very effective if used in loops such as **FOR i=1 TO 32: RANDOMIZE USR routine_address: NEXT i**

The inverter changes the screen attributes without affecting the graphics or text present on the screen and the wrap around routine scrolls the screen lines to the left by a character space, the left hand column being wrapped around and reappearing in the right hand column

```
  1 REM INVERTER BY
         JOHN P.FRYC
  5 CLEAR 40000: INPUT "Enter s
torage address for        INVERTE
R code  ";start
 10 FOR x=start TO start+18: RE
AD a: POKE x,a: NEXT x
 15 SAVE "inverter"CODE start,1
9
 20 DATA 33,0,64,6,24,197,6,0,1
26,238,255,119,35,16,249,193,16,
243,201
```

```
  1 REM wrap around by
          John Fryc.
  5 CLEAR 40000: INPUT "Enter s
torage address for        WRAP AR
OUND code  ";start
 10 FOR x=start TO start+24: RE
AD a: POKE x,a: NEXT x
 15 SAVE "w/around"CODE start,2
5
 20 DATA 6,192,17,0,64,213,225,
35,197,1,31,0,26,237,176,43,119,
0,35,35,19,193,16,240,201
```

Pattern generator

I really couldn't resist this colourful and hypnotic program from Mr. D. McMillan of Glasgow.

The only way to use a program like this is to enter it and experiment, it's fascinating!

Try entering the following: OVER Y, BRIGHT Y, FLASH N, SHIMMER N, INK 7, STEP 2, SIDES 4, OKAY Y.

```
  1 FOR F=32768 TO 32782: READ
J: POKE F,J: NEXT F
 10 FLASH 0: DIM A(8): POKE 236
58,8: PAPER 0: BORDER 0: INK 7:
CLS
 20 RESTORE 300: FOR I=1 TO 8:
READ A$: PRINT A$;
 30 PAUSE 0: PRINT INKEY$
 40 LET A(I)=0: IF INKEY$="Y" T
HEN  LET A(I)=1
 42 IF INKEY$<>"Y" AND INKEY$<>
"N" THEN  LET A(I)=VAL INKEY$
 45 PAUSE 20
 50 NEXT I
 60 IF A(8)=0 THEN  GO TO 20
 70 OVER A(1): BRIGHT A(2): FLA
SH A(3): INK 0+RND*A(5): CLS
 80 LET X=14+INT (RND*60): LET
Z=-A(6)
 90 FOR R=170 TO x STEP z
100 RESTORE A(7)+200: PLOT r,r
110 FOR J=1 TO A(7): READ X1,Y1
: DRAW X1,Y1: NEXT J
120 INK INT (1+RND*A(5))
125 IF A(4)=1 THEN  LET X=USR 3
2768
130 NEXT R
140 GO TO 80
200 DATA 33,0,64,126,203,15,167
,119,35,62,88,188,200,24,244
204 DATA 0,175-2*R,255-2*R,0,0,
-175+2*R,-255+2*R,0
206 DATA 40,.5*(175-2*R),-40,.5
*(175-2*R),255-2*R,0,-40,.5*(-17
5+2*R),40,.5*(-175+2*R),-255+2*R
,0
207 DATA 40,.5*(175-2*R),-40,.5
*(175-2*R),.5*(255-2*R),15,.5*(2
55-2*R),-15,-40,.5*(-175+2*R),40
,.5*(-175+2*R),-255+2*R,0
208 DATA 40,.5*(175-2*R),-40,.5
*(175-2*R),.5*(255-2*R),15,.5*(2
55-2*R),-15,-40,.5*(-175+2*R),40
,.5*(-175+2*R),.5*(-255+2*R),-15
,.5*(-255+2*R),15
300 DATA "OVER (Y OR N)? "
301 DATA "BRIGHT (Y OR N)? "
302 DATA "FLASH (Y OR N)? "
303 DATA "SHIMMER (Y OR N)? "
304 DATA "INK (0-7)? "
305 DATA "STEP (1-9)? "
306 DATA "No. OF SIDES (1-8)? "
307 DATA "OKAY (Y OR N)? "
```

Interface 1/ M-1009 copy

Leif Kornstaedt sent us this routine to produce screen copies on the Brother M-1009 (GLP 2) via the Sinclair Interface 1 RS232 port.

Don't forget to open the "b" channel before calling it, and it may well work on other printers. It isn't very long so it might be worth typing it in to give it a try.

```
  1 REM Interface 1/M1009 copy.
 10 CLEAR 57343: LET f=0
 20 FOR i=1 TO 211: READ a: POK
E 57344+i,a: LET f=f+a: NEXT i
 30 IF f<>26964 THEN  PRINT "ER
ROR - CHECK DATA": STOP
 40 PRINT "To use the copy type
:"''"OPEN #3;""b"":RANDOMIZE USR
 57344: CLOSE #3"
 50 STOP
100 DATA 62,3,205,1,22,62,27,21
5
110 DATA 62,64,215,175,50,212,2
24,175
120 DATA 50,213,224,58,212,224,
79,6
130 DATA 23,129,16,253,71,58,21
3,224
140 DATA 128,50,217,224,33,220,
224,34
150 DATA 218,224,175,50,214,224
,58,217
160 DATA 224,71,62,191,144,50,2
16,224
170 DATA 205,179,224,50,211,224
,62,7
180 DATA 50,215,224,205,201,224
,71,58
190 DATA 211,224,135,128,50,211
,224,58
200 DATA 215,224,61,32,235,42,2
18,224
210 DATA 58,211,224,119,35,34,2
18,224
220 DATA 58,214,224,60,32,197,6
2,27
230 DATA 215,62,42,215,62,3,215
,175
240 DATA 215,62,1,215,33,220,22
4,6
250 DATA 0,126,215,35,16,251,62
,27
260 DATA 215,62,51,215,62,1,215
,62
270 DATA 10,215,62,13,215,58,21
3,224
280 DATA 60,254,3,194,16,224,62
,27
290 DATA 215,62,51,215,62,21,21
5,62
300 DATA 10,215,62,13,215,58,21
2,224
310 DATA 60,254,8,194,12,224,62
,2
320 DATA 195,1,22,58,214,224,79
,58
330 DATA 216,224,71,62,191,205,
172,34
340 DATA 71,4,126,7,16,253,230,
1
350 DATA 201,58,216,224,214,3,5
0,216
360 DATA 224,24,224
```

Paper change

P. Kecskemety provides us with a means of changing the screen paper colour without disturbing the ink, text or graphics.

The code is located in the printer buffer but in fact you can put it anywhere that suits you as it is fully relocatable. Simply use it by entering the command directly or as part of a line — **PRINT USR 23296,x;**

The address 23296 may be changed if you have relocated the routine to another address to that number and the 'x' is any paper colour 0 to 7 that you desire. The final semi-colon is needed to keep the present print position.

```
  10 RESTORE
  20 FOR F=23296 TO 23358
  30 READ E$
  50 IF E$(1)>"9" THEN  LET E=16
*(CODE E$(1)-55): GO TO 70
  60 LET E=VAL E$(1)*16
  70 IF E$(2)>"9" THEN  LET E=E+
CODE E$(2)-55: GO TO 90
  80 LET E=E+VAL E$(2)
  90 POKE F,E
 100 NEXT F
1000 DATA "21","0C","00","39","3
6","4D","23","36","20","E7","CD"
,"FB","24","CD","14","23","FE","
08","D2","44","22","A7","CB","17
","CB","17","CB","17"
1010 DATA "21","00","58","06","1
8","0E","20","5E","CB","9B","CB"
,"A3","CB","AB","B3","77","23","
0D","20","F3","10","EF"
1020 DATA "21","8D","5C","5E","C
B","9B","CB","A3","CB","AB","B3"
,"77","C9"
```

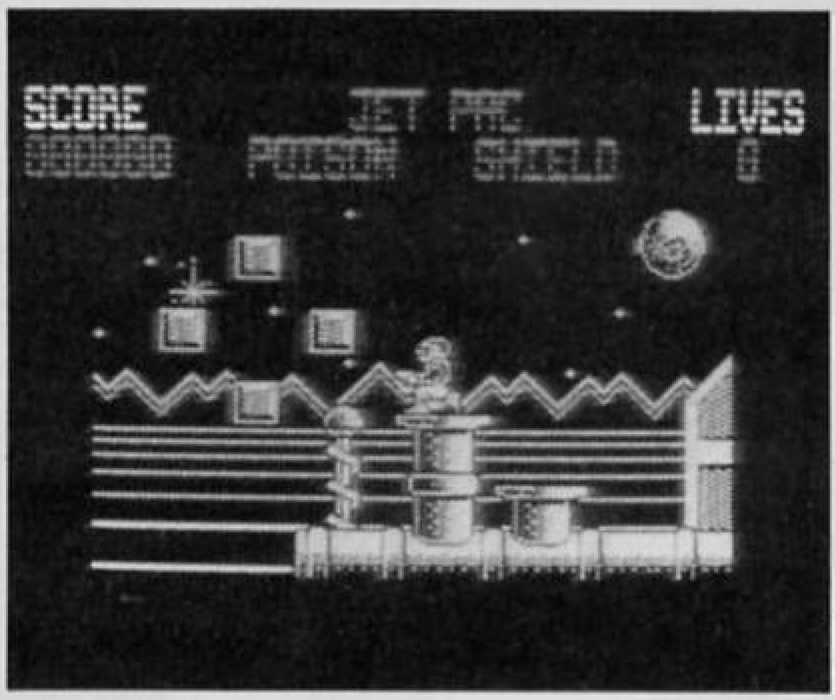
Threads

Can be used as an interesting pattern program in its own right or, as David King the author intended, as a designer program for the popular thread pictures or designs that you create on wood with pins and thread.

Again it is a program which invites experimentation and some intricate and beautiful designs can be developed.

```
 10 LET m=5.5: LET j=44/49: LET
 x$="j*d*COS z+127": LET y$="d*S
IN z+87": PAPER 0: INK 7: BRIGHT
 1: BORDER 0: CLS : GO TO 90
 20 LET z=p
 30 LET x=VAL x$: LET y=VAL y$:
 RETURN
 40 LET z=p+s*k: GO SUB 30
 50 LET u=x-PEEK 23677: LET v=y
-PEEK 23678: IF NOT q THEN  LET
w=u/j: LET t=t+SQR (ABS (w*w)+AB
S (v*v))
 60 RETURN
 70 INPUT "START NEW BOARD? (y/
n) "; LINE s$: IF s$="y" THEN  C
LS
 80 INPUT "Use same pins? (y/n)
 "; LINE p$: IF p$="y" THEN  GO
TO 100+(s$="n")
 90 INPUT "Width? (m for max 5.
5 INS) ";d: LET d=d*87/m: INPUT
"How many pins? ";n: LET s=2*PI/
n
100 FOR p=0 TO 1.99*PI STEP s:
GO SUB 20: PLOT x,y: BEEP .004,0
: NEXT p
110 INPUT "How many pins ON eac
h time? ";k: INPUT "Continuous o
r return? (c/r) "; LINE c$: IF c
$="r" THEN  INPUT "Draw return t
hread? (y/n) "; LINE r$
120 LET z=0: LET t=z: GO SUB 30
: PLOT x,y
130 FOR q=0 TO 2*PI STEP s: GO
SUB 40: DRAW u,v: IF c$="c" THEN
  LET p=z: GO TO 160
140 LET p=p+s: GO SUB 20: GO SU
B 50: IF r$="y" THEN  DRAW u,v:
GO TO 160
150 PLOT x,y
160 NEXT q: PRINT #0;"Thread us
ed ";INT (m/1.75*t*n)/100;" INS"
;TAB 23;"Press Key": PAUSE 0: GO
TO 70
```





WIN GUNRUNNER

There are 25 copies of Hewson's latest game to be won

Gunrunner is a sophisticated shoot'em up set on a planet in the grip of an alien invasion. Your task is guide Gunrunner over a system of plutonium pipes to collect the plutonium which will enable the denizens of the planet to see off the invaders.

In your path are wave upon wave of potentially lethal hardware which obliterates Gunrunner on impact. There are ten levels of action in a game that will test your reflexes to the limit.

We are offering 25 prizes of copies of the game and Gunrunner posters to the winners of this months competition.

How to enter

All you have to do is ring the eight words hidden in the word square. The words can be either horizonal, vertical or diagonal. The words all have some connection with space. When you've done that just fill in the rest of the coupon.

Send your entries to Gunrunners Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.

The Competition is open to all ZX readers except employees of Argus Specialist Publication, Chase Webb and Hewson.

The editors decision is final and no correspondence can be entered into.

The closing date is 10th July 1987.

L	L	A	B	Y	C	M	T	V	Y
X	I	M	V	K	Y	O	O	N	T
T	K	E	L	X	S	V	M	O	L
S	A	T	E	L	L	I	T	E	N
M	O	E	V	S	U	N	O	D	T
S	K	O	T	T	N	V	K	Y	G
L	X	R	G	A	S	N	R	S	E
A	S	T	E	R	O	I	D	R	D
G	B	L	A	C	K	H	O	L	E
I	L	D	M	B	O	V	S	I	N

GUNRUNNER COMPETITION

Name ....

Address ....

....

Cut out this coupon and send to Gunrunner Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.

24 HOUR COMPUTER REPAIRS and COMPUTER SPARES

SINCLAIR
QUALITY APPROVED REPAIR CENTRE

AMSTRAD-SINCLAIR • AMSTRAD APPROVED REPAIR CENTRE • SEAL OF QUALITY •

HOW TO GET YOUR SPECTRUM REPAIRED FOR ONLY £19.95



MANCHESTER CITY CENTRE BRANCH NOW OPEN. CALL FOR DETAILS

WHILE YOU WAIT SERVICE CALL FOR APPOINTMENT

SPECTRUM (only) KEYBOARD REPAIRS £8.95 THE CHEAPEST AROUND

THE NO. 1 REPAIR CENTRE IN THE U.K. OTHERS FOLLOW

WHY NOT COME AND VISIT US AT OUR NEW 2,500 SQ. FT. WORKSHOP. YOU CAN EVEN HAVE A CUP OF COFFEE WHILE YOU WAIT

SPECIAL OFFER!
Why not upgrade your ordinary Spectrum into the fantastic DK'Tronics typewriter keyboard for only **£31.50** including fitting, VAT and return post and packing. Normal recommended retail price **£49.95**. Replacement printed DK'Tronics key sets **£7.50** including post & packing.

SPECIAL OFFER!
Why not upgrade your 16K Spectrum to a 48K for as little as **£17.95** including VAT, post and packing. If you would like us to fit the kit for you just send us **£19.95** which is a fully inclusive price to cover all costs including return postage. Full fitting instructions supplied with every kit. Issue 2 and 3 only.

16K to 48K UPGRADE KIT

Update Your Rubber Keyboard to a New Spectrum+
Fitted for only **£31.90** + £1.50 post & packing (Also D.I.Y. kit available for only **£24.95** + £1.50 post & packing)
Your Spectrum repaired and upgraded to a Spectrum Plus for special offer price of **£50.00** complete. LIMITED OFFER
Same day service

ZX81 SPARES NOW IN STOCK

EXTENSION RIBBON

KEYBOARD TEMPLATES **£5.50** plus £1.50 p & p

TEN ★ REPAIR SERVICE

- ★ While you wait service including computer spare parts over the counter.
- ★ All computers fully overhauled and fully tested before return.
- ★ Fully insured for the return journey.
- ★ Fixed low price of **£19.95** including post, packing and VAT. (Not a between price of really up to **£30.00** which some of our competitors are quoting).
- ★ Discounts for schools and colleges.
- ★ Five top games worth **£39.00** for you to enjoy and play with every Spectrum repair
- ★ We repair Commodore 64's, Vic 20's, Commodore 16's and Plus 4's.
- ★ The most up to date test equipment developed by us to fully test and find all faults within your computer.
- ★ Keyboard repairs, Spectrum rubber key boards only **£8.95**.
- ★ 3 month written guarantee on all repairs.

TRADE REPAIRS most welcome

BEST SERVICE – BEST PRICES!

RUBBER MAT **£6.50** plus £1.50 p & p

56 Way ribbon cable to extend your ports for your peripherals **£10.95** plus £1.50 p & p

REPLACEMENT POWER SUPPLY
Spectrum replacement power transformer suitable for all makes of computer £9.95 plus £1.50 p & p

NEW

TRADE ORDERS WELCOME

Gun Shot II Rapid Fire Joystick and Interface complete outfit. Normal recommended retail price £19.95 if purchased together. Special offer of only **£16.50** including p & p
Joystick available as separate item **£10.95** plus interface available as separate item **£9.00** plus £1.50 p & p

KEYBOARD MEMBRANES
Spectrum keyboard membranes **£5.50** plus £1.50 p & p
Spectrum Plus spare keyboard membranes **£12.90** plus £1.50 p & p
ZX 81 membranes **£5.00** plus £1.50 p & p

ARE YOU ANOTHER CUSTOMER - fed up waiting weeks for your estimate?

Need your computer repaired fast? Then send it now to the **Number One Repair Company in the U.K.**, or call in and see us at our fully equipped 2,500 square foot workshop, with all the **latest test equipment available**. You are more than welcome. We will repair your computer while you wait and help you with any of your technical problems. Commodore computers repaired for only **£35.00**. Please note we gove you a 100% **low fixed price of £19.95** which includes return post and packing, VAT, not a between price like some other Repair Companies offer. We don't ask you to send a cheque in for the maximum amount and **shock you** with repair bills **£30** upwards. Don't forget we are Amstrad aproved for quality and speed, don't risk your computer to any other unauthorised repair centre. We dont just repair the fault and send your computer back, we give your computer a:-

FREE OVERHAUL WITH EVERY REPAIR WE DO:-

We correct **Colour, sound, Keyboard, Check the loading and saving chip**, Put new feet on the base if required, Check for full memory, check all sockets including ear/mike and **replace where needed. All for an inclusive price of £19.95 including VAT, all parts, insurance and post and packing. No hidden extras whatsoever.** We don't have to boast too much about our service as we have thousands of customers from all over the world highly delighted with our service. A **first class reputation for speed and accuracy.** Don't forget, we also now have a Service Branch in **Manchester City Centre** for while you wait service.

COMMODORE POWER SUPPLY
Commodore replacement power transformer £29.00 plus £1.50 p & p

ON-OFF SWITCHES
FOR THE SPECTRUM AND SPECTRUM +
ORDER No.
1067 SPECTRUM
1067A SPECTRUM+
£4.95 + 1.50 p + p

USE ANY JOYSTICK WITH YOUR SPECTRUM + 2
SINCLAIR JOYSTICK ADAPTOR FOR CONNECTING SPECTRUM +2 COMPUTERS TO MOST JOYSTICKS

ORDER No 1052 SELECTOR CONNECTOR R.R.P. £5.95 Our Price £3.50 + 1.50 p+p

THE VIDEO VAULT COMPUTER COMPANION

The Videovault has just released the **most up to date and exciting** catalogue ever produced, our team of experts have scoured the globe to find you all the latest software and add on's for your computer, **all the latest** in joystick technology, all the top and tested **software** titles all at **great discounts** to you. Anyone can apply now, **just send 2 x 18p stamps** to the Catalogue Dept. at the address below, for your **free copy**. We have all the top branded items available, delivery by **return post** (our usual standard). All goods carry our guarantee of satisfaction. Never has such a catalogue ever been produced before, including spare parts, ULA, membranes, power supplies – all at discount prices.

POST + PACKING CHARGES UK ONLY
OVERSEAS POST + PACKING CHARGES PLEASE WRITE FOR QUOTE

FULLY REPAIRED AND TESTED ON MOST ADVANCED TEST EQUIPMENT IN EUROPE!

VideoVault Ltd.

140 High Street West, Glossop, Derbyshire SK13 8HJ
Tel: 04574-66555/67761 Head office & access orders, queeries,
Manchester 061- 236 0376 while you wait repair centre only.

OPEN 7 DAYS A WEEK
MANCHESTER MON - SAT ONLY

NOW SPECIAL OFFER
6 GREAT FREE GAMES
NORMAL RECOMMENDED RETAIL PRICE £39.00
YOURS FREE WITH EVERY SPECTRUM REPAIR

WE NOW HAVE FAX: (0) 4574 68946

PROJECT X

Some of the many solutions to the second problem we posed.

They poured in from all over the place, long, short, complex and simple. I locked myself away for a couple of days and went through them all.

The solutions I am featuring are not chosen as necessarily being the best, but those which demonstrate as wide a range of methods as possible.

♠♦♥♣



Chris Oliver used a method where the card data is stored in a first line REM and then PEEKed at (memories of my ZX81 days). It may be better to amend line 10 so that the variable 'table' is set up by **PEEK 23635+256*PEEK 23636+5** to allow for such peripherals as the interface 1 and microdrives which will alter that address.

There is also a chance that picking cards using a simple RND function will cause some repetitiveness, using a continual GOSUB while waiting for a key-press ensures a more efficient randomness. Chris kept the display to a minimum but enough to show that the routines are effective.

```
   1 REM A23456789TJQKA23456789T
JQKA23456789TJOKA23450789TJQK
   2 REM Oliver
  10 LET table=23759: GO SUB 500
 : RANDOMIZE
  20 BORDER 4: PAPER 4: INK 0: C
LS : GO SUB 400
  30 LET row=3: LET col=6
  40 CLS : GO SUB 100: GO SUB 20
0
  50 LET col=col+3: GO SUB 300:
GO SUB 120: GO SUB 200
  90 STOP
  98
  99 REM  DEAL A CARD ROUTINE
 100 PRINT #1;"Shuffling the car
ds !"' FLASH 1;"Press any key to
 deal.": FLASH 0
 110 GO SUB 300: IF INKEY$="" TH
EN  GO TO 110
 120 LET p$=CHR$ PEEK (table+a)
 130 IF p$="0" THEN  GO SUB 300:
 GO TO 120
 140 LET s$=CHR$ (144+INT ((a-1)
/13))
 150 LET ink=2-(2*(a>26))
 160 POKE (table+a),48: RETURN
 198
 199 REM  DISPLAY ROUTINE
 200 PRINT  PAPER 7; INK ink;AT
row,col;p$;" ";AT row+1,col;" ";
s$: RETURN
 298
 299 REM  CREATE A RANDOM NUMBER
 300 LET a=1+INT (RND*52): RETUR
N
 398
 399 REM  DATA FOR SUIT GRAPHIC
SYMBOLS
 400 FOR f=USR "a" TO USR "d"+7:
 READ d: POKE f,d: NEXT f: RETUR
N
 410 DATA 34,119,127,127,62,62,2
8,8,8,28,62,127,62,28,8,0
 420 DATA 28,28,8,107,127,107,8,
8,8,28,62,127,127,127,107,8
 498
 499 REM  INITIALISE PACK
 500 LET a$="A23456789TJQK"
 510 FOR k=0 TO 39 STEP 13: FOR
f=1 TO 13: POKE (table+k+f),CODE
 a$(f): NEXT f
 520 NEXT k: RETURN
```

Don Menary sent us two routines in which the displays differed, we used the one which did the task. He kept track of which cards had been previously used by setting the array a$ to a * once selected.

```
  0>REM Don Menary MARCH 87
   5 BORDER 7: PAPER 7: INK 0: C
LS
  10 FOR g=USR "a" TO USR "i"+7
  20 READ a: POKE g,a
  30 NEXT g
  40 DATA 36,110,247,255,255,126
,60,24
  41 DATA 24,60,126,255,255,247,
110,36
  42 DATA 24,60,126,255,255,126,
60,24
  43 DATA 24,60,126,255,255,126,
60,24
  45 DATA 24,60,60,255,255,255,9
0,24
  46 DATA 24,90,255,255,255,60,6
0,24
  47 DATA 24,60,126,255,255,255,
90,24
  48 DATA 24,90,255,255,255,126,
60,24
  49 DATA 0,78,81,81,81,81,78,0
  55 LET P$="123456789TJQK123456
789TJQK123456789TJQK123456789TJQ
K"
  60 FOR i=103 TO 200 STEP 96
  61 PLOT i,40
  62 DRAW 8,8,PI/4
  63 DRAW 0,87
  64 DRAW -8,8,PI/4
  65 DRAW -55,0
  66 DRAW -8,-8,PI/4
  67 DRAW 0,-87
  68 DRAW 8,-8,PI/4
  69 DRAW 56,0
  70 NEXT i
 100 DIM a$(52)
 110 FOR g=1 TO 52 STEP 2
 120 FOR f=1 TO 2
 130 LET a=INT (RND*52)+1
 140 IF a$(a)<>" " THEN  GO TO 1
30
 150 LET a$(a)="*"
 160 LET i=NOT i
 170 GO SUB 210
 180 NEXT f
 190 PRINT #0;AT 0,0;"PRESS ANY
KEY FOR NEXT TWO CARDS": PAUSE 0
: INPUT ""
 192 FOR d=5 TO 15
 194 PRINT AT d,i+5;"          ";AT
 d,i+17;"          "
 196 NEXT d
 198 NEXT g
 200 PRINT #0;AT 0,3;"THATS ALL
52 CARDS DEALT": PAUSE 0: INPUT
"": GO TO 100
 220 LET c$=("E" AND a<14)+("G"
AND a>13 AND a<27)+("C" AND a>26
 AND a<40)+("A" AND a>39)
 225 INK (0 AND a<27)
 230 INK (2 AND a>26)
 235 LET col=i+6+(i*11)
 240 PRINT AT 5,col;(p$(a) AND p
$(a)<>"T" AND p$(a)<>"1")+("I" A
ND p$(a)="T")+("A" AND p$(a)="1"
)
 245 PRINT AT 6,col;c$;AT 14,col
+6;CHR$ (CODE c$+1)
 247 IF p$(a)="J" OR p$(a)="Q" O
R p$(a)="K" THEN  PRINT AT 10,co
l+3;p$(a): GO TO 280
 250 IF p$(a)="1" OR p$(a)="3" O
R p$(a)="5" OR p$(a)="9" THEN  P
RINT AT 10,col+3;c$
 255 IF p$(a)="7" OR p$(a)="8" O
R p$(a)="T" THEN  PRINT AT 8,col
+3;c$
 257 IF p$(a)="2" OR p$(a)="3" T
HEN  PRINT AT 7,col+3;c$
 260 IF p$(a)="8" OR p$(a)="T" T
HEN  PRINT AT 12,col+3;CHR$ (COD
E c$+1)
 262 IF p$(a)="2" OR p$(a)="3" T
HEN  PRINT AT 13,col+3;CHR$ (COD
E c$+1)
```

```
 265 IF p$(a)>"3" THEN  PRINT AT
 7,col+1;c$;AT 7,col+5;c$;AT 13,
col+1;CHR$ (CODE c$+1);AT 13,col
+5;CHR$ (CODE c$+1)
 270 IF p$(a)>"5" AND p$(a)<"9"
THEN  PRINT AT 10,col+1;c$;AT 10
,col+5;c$
 275 IF p$(a)>"8" THEN  PRINT AT
 11,col+1;CHR$ (CODE c$+1);AT 11
,col+5;CHR$ (CODE c$+1);AT 9,col
+1;c$;AT 9,col+5;c$
 300 FOR u=0 TO 8
 310 FOR y=1 TO 8
 320 IF POINT (col*8+u,175-(6*8)
+y)=1 THEN  PLOT -(col+7)*8+u,17
5-(15*8+y)
 330 NEXT y
 340 NEXT u
 400 RETURN
 500 SAVE "Card Deal" LINE 1
```

David Reed was the only reader to dispense with the need for UDGs and use the PLOT and DRAW functions. The problem is the slowness of the drawing, perhaps using UDGs in conjunction with the graphics method, possibly for the suit shapes, might overcome this delay. It's an interesting attempt though.

```
  0>REM PROJECTX by D.REED
 10 RANDOMIZE : REM  3/87
 20 GO SUB 200: REM Colours
 30 GO SUB 220: REM Variables
 40 GO SUB 250: REM Choose card
 50 GO SUB 290: REM Draw card
 60 GO SUB 350: REM Mark card
 70 GO SUB 90 : REM Scroll card
 80 GO SUB 250: GO SUB 290: GO
SUB 350: INPUT "   Press enter t
o run again "; LINE a$: BEEP .01
,38: GO TO 20
 90 POKE 23692,255: FOR C=1 TO
11: PRINT AT 21,31,,: NEXT C: RE
TURN
 100 DATA 2,0,-5,-8,5,0,8,5,0
 110 DATA 4,4.4,.4,4,1,1,0,-1.4,
0,0,-1.4,0,-4,1,1
 120 DATA 4,0,5,-2,-4,1.5,-6,-1,
3,6,-1,3,2,-4,1.5
 130 DATA 2,0,5,8,-5,0,-8,-5,0
 140 DATA 4,0,5,5,-3,-2,-2,-2,-1
 5,2,-2,-1.5,-5,-3,-2
 150 DATA 100,130,0
 160 DATA 100,140,0
 170 DATA 120,140,110
 180 DATA 100,140,110
 190 PLOT  PAPER K; INVERSE 1;I,
J: DRAW  INVERSE 1; PAPER K;1,0:
 RETURN
 200 PAPER 0: INK 7: INVERSE 0
 210 BORDER 0: CLS : RETURN
 220 DIM A(52): FOR B=1 TO 52
 230 LET A(B)=B: NEXT B: DIM M(3
)
 240 LET B=52: LET A$="JQK": RET
URN
 250 LET C=INT (RND*B)+1
 260 LET D=A(C): LET A(C)=A(B)
 270 LET B=B-1: LET E=INT ((D-1)
/13)
 280 LET F=D-(E*13): RETURN
 290 FOR C=4 TO 75: PLOT 52,C
 300 DRAW 151,0: NEXT C
 310 NEXT C: FOR C=0 TO 3
 320 PLOT 55-C,C: LET G=C*2
 330 DRAW 145+G,0: PLOT 52+C,76+
C
 340 DRAW 150-G,0: NEXT C: RETUR
N
 350 LET G=8: PRINT  INK 0;AT 21
,0;"A": IF F>1 THEN  LET G=(LEN
STR$ F)*8: PRINT  INK 0;AT 21,0;
F: IF F>10 THEN  PRINT  INK 0;AT
 21,0;A$(F-10);"  ": LET G=8
 360 LET L=-1: LET K=0: IF E<2 T
HEN  LET K=2: LET L=1
 370 FOR C=0 TO G: FOR D=0 TO 7
 380 LET H=D*2: IF POINT (C,D) T
HEN  LET I=70-H: LET J=C+4: GO S
UB 190: LET J=J+72-G: GO SUB 190
: LET I=H+183: LET J=75-C: GO SU
B 190: LET J=3+G-C: GO SUB 190
 390 NEXT D: NEXT C: PRINT AT 21
,0;"  ": LET N=127: LET O=40
 400 LET D=1: IF F>10 THEN  LET
D=4: GO SUB 500: RETURN
 410 LET T=F: LET U=0: IF F<4 TH
EN  LET P=2: LET Q=F: GO SUB 460
: RETURN
 420 IF T<2 THEN  GO TO 440
 430 IF (U/2)-1<T THEN  LET T=T-
2: LET U=U+2: GO TO 420
 440 LET P=1: LET Q=U/2: GO SUB
450: LET P=3: GO SUB 450: LET P=
2: LET Q=T: GO SUB 450: RETURN
 339 IF Q<1 THEN  RETURN
 460 LET W=INT (Q/2): LET V=L: L
ET L=-L
 470 LET O=20*P: LET R=INT (130/
(Q+1)): FOR S=1 TO Q: LET N=(S*R
)+62
 480 IF S>W THEN  LET L=V
 490 GO SUB 500: NEXT S: RETURN
 500 RESTORE 150+(E*10): FOR C=1
 TO 3: READ M(C): NEXT C
 510 FOR C=1 TO 3: LET H=M(C): I
F NOT H THEN  GO TO 530
 520 RESTORE H: READ G,H,I: PLOT
  INVERSE 1;((H*D)*L)+N,(I*D)+O:
 FOR G=1 TO G: READ H,I,J: DRAW
 INVERSE 1;(H*D)*L,(I*D),J*L: NE
XT G: NEXT C
 530 RETURN
```

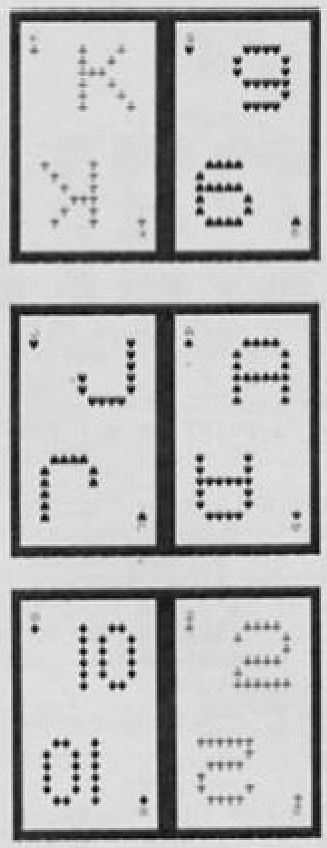
S. M. Goodman's is typical of the short, efficient routines we received which simply defined UDGs as suit shapes and concentrated on the shuffling. In fact the shuffle seemed to present no problems to anyone who wrote in, the biggest difference seemed to be in the display methods.

Also along the same lines were routines from V. F. Hey and old friend John E. Amphlett.

```
  1 REM S.M.Goodman - RUN 90
        then just RUN
 10 LET a$="AB2B3B4B5B6B7B8B9BF
BJBQBKBAC2C3C4C5C6C7C8C9CFCJCQCK
CAD2D3D4D5D6D7D8D9DFDJDQDKDAE2E3
E4E5E6E7E8E9EFEJEQEKE"
 20 FOR n=1 TO 4
 40 LET x=INT (RND*103)+1: IF x
/2=INT (x/2) OR a$(x)="0" THEN
GO TO 40
 60 LET j$=a$(x+1): IF j$="D" O
R j$="E" THEN  PRINT  INK 2;a$(x
);j$,
 65 IF j$="B" OR j$="C" THEN  P
RINT a$(x);j$,
 70 LET a$(x)="0": NEXT n: STOP

 90 FOR q=USR "b" TO USR "b"+39
: READ g: POKE q,g: NEXT q: RUN
 100 DATA 56,56,254,254,214,16,5
6,0,16,56,124,254,254,16,56,0,16
,56,124,254,124,56,16,0,66,231,2
55,255,126,60,24,0,0,156,162,162
,162,162,156,0
```

W. E. Thomson abandoned the conventional card display and created an effective alternative of his own, where each card number or letter is created in large form made out of the suit UDG's and mirrored. This removes the need for creating the complex court cards and so keeps the program size down.



```
  5 REM  W.E.Thomson
  9 REM ***PROLOGUE***
 10 BORDER 4
 20 PRINT "Please wait half a m
inute       while I produce 21 U
DGs"
 30 GO SUB 1000
 40 LET A=1: LET T=10: LET J=11
: LET Q=12: LET K=13
 50 LET C=1: LET D=2: LET H=3:
LET S=4
 60 RANDOMIZE
 99 REM ***SHUFFLE***
 100 DIM p$(104): LET p$="ASKSQS
JSTS9S8S7S6S5S4S3S2SAHKHQHJHTH9H
8H7H6H5H4H3H2HADKDQDJDTD9D8D7D6D
5D4D3D2DACKCQCJCTC9C8C7C6C5C4C3C
2C"
 110 CLS : PRINT "Shuffled pack
is:"
 120 FOR m=52 TO 1 STEP -1
 130 LET n=2*INT (m*RND)+1
 140 PRINT p$(n TO n+1);
 150 LET p$=p$( TO n-1)+p$(n+2 T
O )+p$(n TO n+1): NEXT m
 199 REM *** EXAMINE SHUFFLE***
 200 PRINT '"Press any key to di
splay       the 1st two cards":
 PAUSE 0
 299 REM ***PRINT BACKGROUND***
 300 CLS : INK 4: FOR r=0 TO 21
 310 FOR p=0 TO 31
 320 IF 0<r AND r<21 THEN  IF p=
1 OR p=17 THEN  LET p=p+14
 330 PRINT AT r,p;"■"
 340 NEXT p: NEXT r: INK 0
 399 REM ***DISPLAY 2 CARDS***
 400 FOR n=0 TO 1
 410 LET z$=p$(2*n+1)
 420 IF z$<>"T" THEN  PRINT AT 2
,2+16*n;z$: GO TO 470
 430 RESTORE 460
 440 FOR w=16706+16*n TO 17986+1
6*n STEP 256
 450 READ b: POKE w,b: NEXT w
 460 DATA 76,82,82,82,82,76
 470 LET y$=p$(2*n+2)
 480 LET nk=2 AND (y$="D" OR y$=
"H")
 490 LET c$=CHR$ (156+VAL y$)
 500 PRINT AT 3,2+16*n; INK nk;c
$
 510 LET r=3: LET c0=7+16*n: LET
 x0=17+128*n: LET y=159
 520 GO SUB 700
 530 PRINT AT 19,13+16*n;CHR$ (1
43+VAL z$)
 540 LET c$=CHR$ (160+VAL y$)
 550 PRINT AT 18,13+16*n; INK nk
;c$
```

▶

```
560 LET r=13: LET c0=c0-4: LET
x0=x0+88: LET y=23
570 GO SUB 700
580 NEXT n
599 REM ***MORE?***
600 PRINT #0;"Press 'r' to resh
uffle,          's' to  STOP "
610 IF INKEY$="" THEN  GO TO 61
0
620 IF INKEY$="r" THEN  GO TO 1
10
630 STOP
699 REM ***CARD VALUE (BIG)***
700 FOR r=r TO r+5: LET y=y-1
710 PRINT AT r,c0;
720 FOR x=x0 TO x0+5
730 LET z=POINT (x,y): PRINT  I
NK nk;(c$ AND z);(" " AND NOT z)
;
740 NEXT x: NEXT r: RETURN
999 REM ***PRODUCE UDGs***
1000 LET s$="A23456789TJQK"
1010 RESTORE 1100: FOR i=1 TO 13
1020 PRINT "*";: IF i=10 THEN  F
OR j=0 TO 7: READ q: GO TO 1080
1030 LET z=15367+8*CODE s$(i)
1040 FOR j=0 TO 7
1050 LET p=PEEK (z-j): LET q=0
1060 FOR k=0 TO 7
1070 LET r=INT (p/2): LET q=2*(q
-r)+p: LET p=r: NEXT k
1080 POKE USR CHR$ (143+i)+j,q
1090 NEXT j: NEXT i
1100 DATA 0,50,74,74,74,74,50,0
1110 FOR i=1 TO 4: PRINT "**";
1120 FOR j=0 TO 7
1130 READ b: POKE USR CHR$ (156+
i)+j,b: POKE USR CHR$ (160+i)+7-
j,b/2
1140 NEXT j: NEXT i: RETURN
1150 DATA 16,56,16,84,254,84,16,
16
1160 DATA 16,56,124,254,254,124,
56,16
1170 DATA 68,238,254,254,254,124
,56,16
1180 DATA 16,56,124,254,254,254,
84,16
```

Philip Byford is a man after my own heart, when things get tough — cheat! The actual setting up of the pack was a little unusual in that he used the CHR$ function to set up a string with *chars* representing the cards then decoded using two clever calculations. The graphic representation is where the cheating comes in. Although the cards are realistically layered the top card should be fully drawn. Still, this would have made the program unprintably long.

```
   1 REM Undelined letters are
         UDG's
  10 REM Philip Byford
  20 REM Initialisation
  30 BORDER 0: PAPER 4: INK 7: B
RIGHT 1
  40 GO SUB 620
  50 LET a$=""
  60 FOR n=1 TO 52
  70 LET a$=a$+CHR$ (n)
  80 NEXT n
  90 LET c$="DEFGHIJKLMNOP"
 100 LET d$="QRST"
 110 LET b$=""
 120 REM Shuffle & Deal
 130 CLS
 140 PRINT AT 4,9;"PLEASE WAIT:"
;AT 6,5;"CARDS BEING SHUFFLED"
 150 RANDOMIZE
 160 FOR n=52 TO 2 STEP -1
 170 LET a=INT (RND*n)+1
 180 LET b$=b$+a$(a)
 190 LET a$=a$( TO a-1)+ a$(a+1
 TO )
 200 NEXT n
 210 LET a$=b$+a$
 220 GO SUB 370
 230 FOR n=1 TO b
 240 LET a=CODE a$(n)
 250 LET e=INT ((a-1)/13)
 260 LET c=a-e*13
 270 LET d=e+1
 280 LET f=31-(n-1)*2
 290 PRINT  PAPER 7; INK (2 AND
a<27);AT 11,f;c$(c);AT 12,f;d$(d
)
 300 FOR m=13 TO 20
 310 PRINT  PAPER 7;AT m,f;" "
 320 NEXT m
 330 NEXT n
 340 PAUSE 0
 350 GO TO 110
 360 REM Graphics
 370 PRINT AT 4,8;"DEALING CARDS
:";AT 6,5;"PLEASE WAIT A MOMENT"
 380 INK 4
 390 FOR n=0 TO 30 STEP 2
 400 PRINT AT 10,n;"A";AT 21,n;"
B"
 410 FOR m=11 TO 20
 420 PRINT AT m,n+1;"C"
 430 NEXT m
 440 NEXT n
 450 FOR n=1 TO 241 STEP 16
 460 PLOT n,8
 470 DRAW 0,79
 480 NEXT n
 490 PLOT 8,95
 500 DRAW 247,0
 510 FOR m=0 TO 31
 520 FOR n=10 TO 21
 530 PRINT AT n,m; OVER 1; PAPER
 7; INK 0;" "
 540 NEXT n
 550 NEXT m
 560 INK 7
 570 PRINT AT 4,8;"HOW MANY CARD
S";AT 6,4;"DO YOU WANT TO TURN O
VER?"
 580 INPUT b
 590 PRINT AT 4,8;"YOUR SELECTIO
N:";AT 6,4;"
 "
 600 RETURN

 610 REM User-Defined-Graphics
 620 RESTORE 680
 630 FOR n=0 TO 159
 640 READ d
 650 POKE USR "a"+n,d
 660 NEXT n
 670 RETURN
 680 DATA 0,3,12,16,16,32,32,64
 690 DATA 64,64,32,32,16,16,12,3
 700 DATA 170,213,170,213,170,21
3,170,213
 710 DATA 24,36,66,66,126,66,66,
231
 720 DATA 120,132,4,4,56,64,132,
252
 730 DATA 252,132,8,48,8,4,132,1
20
 740 DATA 8,24,40,72,136,254,8,2
8
 750 DATA 252,128,128,248,4,4,13
2,120
 760 DATA 120,132,128,184,196,13
2,132,120
 770 DATA 252,132,8,16,32,64,64,
64
 780 DATA 120,132,132,120,132,13
2,132,120
 790 DATA 120,132,132,140,116,4,
132,120
 800 DATA 78,209,81,81,81,81,81,
238
 810 DATA 62,8,8,8,8,8,136,112
 820 DATA 120,132,132,132,132,24
4,140,126
 830 DATA 231,66,68,72,88,100,66
,231
 840 DATA 0,108,254,254,254,124,
56,16
 850 DATA 0,16,56,124,254,124,56
,16
 860 DATA 16,56,56,214,254,214,1
6,56
 870 DATA 16,56,124,254,254,214,
16,56
```

Finally, **P. Kecskemety** used an efficient system where cards are selected from the initial A$ and put into C$ (lines 30 to 70), the remainder of the program presents a neat and tidy display.

```
   1 REM kecskemety
   5 RANDOMIZE
   7 GO SUB 9000
  10 DIM C$(104)
  20 LET A$=" 2H3H4H5H6H7H8H9HTH
JHQHKHAH2S3S4S5S6S7S8S9STSJSQSKS
AS2D3D4D5D6D7D8D9DTDJDQDKDAD2C3C
4C5C6C7C8C9CTCJCQCKCAC"
  25 PRINT TAB 2;"SHUFFLING CARD
S. PLEASE WAIT"
  28 REM Shuffle
  30 FOR F=52 TO 1 STEP -1
  40 LET A=INT (RND*F+1)*2
  50 LET C$(F*2-1 TO F*2)=A$(A T
O A+1)
  60 LET A$=A$( TO A-1)+A$(A+2 T
O )
  70 NEXT F
  80 PAPER 4: CLS : INK 1
  90 PRINT  PAPER 7;AT 7,15;"▛▀▀
▜";AT 8,15;"▌XX▐";AT 9,15;"▌XX▐"
;AT 10,15;"▌XX▐";AT 11,15;"▌XX▐"
;AT 12,15;"▙▄▄▟"
  95 INK 0: PLOT 119,118: DRAW -
7,-7: DRAW 0,-47: DRAW 7,7: DRAW
 -7,-7: DRAW 31,0: DRAW 7,7
 100 LET X$=C$( TO 2)
 110 LET Y$=C$(3 TO 4)
 120 LET C$=C$(5 TO )
 130 LET Y=4: LET Z$=X$: GO SUB
1000
 140 LET Y=24: LET Z$=Y$: GO SUB
 1000
 150 GO TO 1E4
1000 INK 0+2*(Z$(2)="D" OR Z$(2)
="H")
1010 PRINT  PAPER 7;AT 8,Y;Z$(1)
;"   ";AT 9,Y;Z$(2);"    ";AT 10,
Y;"    ";AT 11,Y;"    ";AT 12,Y;
"    ";Z$(2);AT 13,Y;"   ";Z$(1)
1020 PLOT  INK 0;Y*8-1,110: DRAW
  INK 0;0,-47: DRAW  INK 0;32,0
1040 RETURN
9000 RESTORE
9010 FOR F=0 TO 7: READ A: POKE
USR "C"+F,A: NEXT F
9020 FOR F=0 TO 7: READ A: POKE
USR "D"+F,A: NEXT F
9030 FOR F=0 TO 7: READ A: POKE
USR "H"+F,A: NEXT F
9040 FOR F=0 TO 7: READ A: POKE
USR "S"+F,A: NEXT F
9050 FOR F=0 TO 7: READ A: POKE
USR "T"+F,A: NEXT F
9060 RETURN
9100 DATA 0,28,28,127,127,119,8,
28
9110 DATA 0,8,28,62,127,62,28,8
9120 DATA 0,119,127,127,127,62,2
8,8
9130 DATA 0,8,28,62,127,119,8,28
9140 DATA 0,38,41,41,41,41,38,0
```

Mentions should also be made of Daniel McIntyre who, at age eleven, made a most creditable attempt, Mr. McMiller (?) who was the only person to attempt to draw the court cards and subsequently his program became too large to use. A pity because it was a good program.

Regulars Ian Rogers and Rene Uittenbogaard along with K. N. Battye, Mr. B. Rumbelow, Dean Hickingbottom, E. F. Taylor and P. Bocij also deserve a mention for impressive programs.

Once again my thanks to everyone who wrote and sent in comments or listings. I'm sorry that I couldn't mention all of you, but there are not enough pages in the mag!

CROSSWIRES

Ray Elder solves your problems.

Q *Dear Sir,*
I own a Spectrum 48K. My father has an Epson LX-80 printer and I would like to know if the printer is compatible with my Speccy.
If so, then which interface and what hardware and software will I need to operate the Epson. Also could you please give me the prices for the items I need.
Samuel MacDonald.

A Yes, your father's LX-80 is compatible provided that you use a Centronics interface. I would suggest that the best ones available are the Kempston E and the ZX Lprint 3, both have all the necessary software built into the interface, are easy to use and are all that you will require to operate the printer. As you live in Norway then I would check on the postage cost from the company.

The ZX Lprint 3 is available from Transform Ltd, 24 West Oak, Beckenham, Kent BR3 2EZ (Tel: 01-658 6350) for £29.95 plus £8.85 for a connecting cable. The Kempston E costs £49.95 and includes the printer cable, they can be contacted at Kempston Electronics (0908-690018).

Q *Dear Sir,*
My joystick/interface has started playing up. At first it was slow to respond to the left and then the 'up' failed to work.
Is it the interface or the joystick and can it be repaired? I have a Quickshot joystick and a Sinclair interface 2.
Robert Davis.

A Hard to say, the best way is to try a friend's joystick with your interface and your joystick with his interface and see which is giving the problem. Of course it could be both!

Q *Dear Sir,*
I hope to purchase either an Amstrad DMP 2000 printer as recommended by yourselves, or maybe an Epson LX-86 to go with my Spectrum 128K. I would be grateful if you would advise me as to which is the interface (or combination of interfaces) most likely to enable me to print screen dumps and word processing results in both 128 and 48K modes.
The reason I ask is that I have your Specword 128 program and the Artist 2 which, although I asked for the 128K version, will only run when in 48K mode. I almost bought the 128K version of the discovery Disk drive being prompted by your Discovery column, but unfortunately the moment I had saved enough to buy one they went out of stock and production. Do you know if there are any plans to reintroduce this machine.
Dean Shepherd.

A Last question first; no, there are no plans (at the moment) to re-introduce the Discovery.

Actually there is one answer to all your problems, but it will cost around £200. One of the major problems has been getting an interface that will operate a printer in both 128 and 48K modes, the existing ones tend to work in one or the other.

The solution is the Disciple combination printer/joystick AND disk drive interface. You can get it for £89.95 from Rockfort Products, 81 Church Road, London NW4 4DP and use the joystick/printer connections then later buy a disk drive to add to it.

I personally wouldn't recommend the LX-86 as its NLQ quality is disappointing. If you want to print graphics then look for a printer with 640 dots per line in Bit Image mode.

Q *Dear Sir,*
I am a newcomer to computing and I recently bought a Spectrum unit and a 128K Spectrum. Could you tell me if it is possible to synchronise the Spectrum with my keyboard, which is a Yamaha MK100 (not MIDI compatible)?
William W. Dewar.

A Bad news William, unless you know an electronic wizard who can build an interface and amend the MK100 internally, you are unable to do this.

I would imagine from your letter that you are a music enthusiast and my personal recommendation is that you try to upgrade your keyboard to a MIDI one. If you do, make sure it has not only the OMNI mode of operation but the MONO mode as well, or else you are likely to suffer further frustration as you develop your interest.

Q *Dear Sir,*
I am requesting assistance with what may seem a simple problem to you. My problem is that I want to create pictures or SCREEN$ and use them either as front loading screens or as game screens. However, whenever I then load in the main code the picture is spoilt by the titles or bytes of the next program being printed across them as it loads.
Is there any way to eliminate this?
A. A. Lewis.

A This is a case of pre-planning both your picture and the loading program. Leave the first half of any line clear, it can be any paper colour but should have no inked lines or text, make a note of the paper colour and line number. Save your main program with a single letter name or even just a space!

In your loader program use a line such as **LOAD "" SCREEN$: PAPER colour: INK colour: PRINT AT line— 1,0: LOAD ""**. Where both the paper and ink numbers are the same as the paper colour of that line. Now your progam title will be hidden when it is loaded.

And finally a few quickies:

Chris Homsy has problems with his twin microdrive system where drive two formats and CATS a particular cartridge perfectly, but drive one won't. I suggest changing the positions of the drives and see if the same drive in position two still fails, if so then perhaps its motor is weaker than the other and cannot cope with a slightly stiff tape. If it does cope then it could be a new Interface one bug!

In the March issue I said I did not know of an interface that would allow connection of the 1541 drive to a 48K Spectrum. Edwin Groof**** (I couldn't read the rest of the name) of Holland says that Logitek, Pankstrasse 49, 1000 Berlin, Germany, make one which also connects to a printer.

Good friend John Wase is in need of a program which will output to a HI-80 printer plotter graphs which are scaled, drawn with named axes and printed with multiple curves and parameters — such as Wychwood for the BBC.

Anyone help?

EXPERT SYSTEMS

Part 5: Question time.

BY DAVID NOWOTNIK

What would be your reaction if you went to your doctor with something which seemed like 'flu, and he asked if you had an ingrown toenail? Even if you don't now much about medicine, the question is obviously rather irrelevant, and (to answer my question) you'd wonder whether your doctor really knew anything about medicine.

Asking an obviously irrelevant is just one way an expert can lose credibility. And the person consulting the expert must have confidence in the expert if he is to believe the advice the expert has to offer.

So far in this series, I have indicated that a computer expert system should adopt a parallel approach to decision making to obtain the most reliable response. In other words (if you haven't been following the series), all questions necessary to form a decision must be asked prior to making any judgements. With a very small (and, so not very practical) knowledge base, most questions asked should be reasonably relevant. After all, it is difficult to go off track with only a handful of outcomes to choose from.

But, any expert system worthy of continued use should contain a relatively large amount of 'knowledge'. Under these circumstances, a fully parallel approach would need to ask a vast amount of questions, many totally irrelevant to the problem in hand.

So, to create an expert system of any magnitude which can demonstrate some intelligence (albeit artificial) in selecting questions which are relevant to the problem, other approaches are necessary. And given that a serial approach can be unreliable (one error in response by the user, and the expert will be wrong), what can be done? I'll now try to supply a few answers to that dilemma, using some comon sense, some probability theory and a different decision making strategy.

Common sense

In trying to solve most problems it is quite often necessary to ask some broad questions to identify in roughly which area the problem exists, or where the solution can be found. This established, the questions can then be quite specific to pinpoint the problem, and supply a solution. This breaks the problem of asking questions down to two (or more) parts; getting the right area sorted out, then homing in on the problem in one or more steps.

In an earlier part of the series, I gave an example of an expert system which advised on computer selection. One fundamental question was whether the user wanted a computer for 'casual' home use, or whether he was a business user. If you assume that the user knows the answer to that, with complete certainty, then an expert system could initially use a serial approach — a question such as 'Will you use a computer for business only?' would separate business from home computer users. Having made that distinction, then one set of questions can be applied to a business user and a quite different set for the home computer user, either one using a parallel approach in coming to the final decision.



```
Table 1. Main variables and DATA structures in the Node
         Expert System Program

a) Main Variables

a$()  Flag to show if a question has already been asked,
      and, if so, the response (y/n).
nd    Current node in use.
node  The total number of nodes.
q$()  Array holding the question strings.
o$()  Array holding the replies (outcomes).
type  The type of node (1=serial, 2=parallel).
t$    The title of the system.

b) Arrangement of DATA statements

i) Lines 5000-5100 - System information

Title of the system
Number of nodes
Number of questions
Number of outcomes

ii) Lines 5500 to end - Node information

Information on node n starts at line (5000+n*500)

For serial nodes, the order of data is as follows:

Node type (1=serial)
The number of questions (qu)
Then for each question in turn:
     the question number [position in q$()]
     Where to go on yes/no reply
        (q=another question, n=another node, o=provide a
         response or outcome)
     question/node/response number on yes
     question/node/response number on no


For parallel nodes:

Node type (2=parallel)
The number of questions (qu)
The question numbers [position in q$()] of qu questions
The number of outcomes (n)
The outcome type for each outcome in turn
     (e.g. n12= node 12, o05=reply 5)
```

What you have in this case is a mixture of parallel and serial decision making. In effect, you have two quite separate expert systems using a parallel approach. Which expert system is used depends upon the response to one question, and a serial decision making process to select the appropriate parallel expert system. In this way, the 'pure' business user need not answer irrelevant questions about interests in, say machine code, and the home user about spreadsheets.

Nodes

This is a simple example of breaking up an expert sytem into smaller decision making units. In this example, the first node (each decision making is called a node) is the serial process to dicriminate between home and business user; thereafter, there are two nodes, both parallel, and both quite independent (i.e. the outcome of one has no influence on the outcome of the other). Fig. 1 shows this simple system in diagram form. Node 1 leads to node 2 if the user of the expert system says he is a business user, or node 3 if he is a home computer user. Nodes 2 and 3 can then be quite independent parallel decision making expert systems specialising in business and home systems respectively. The outcome from either node 2 or 3 is a response (answer) from the expert system.

By dividing up the expert system in this way, we have prevented the system asking a lot of irrelevant questions, but it still has a flaw. The very first question assumes that the user knows whether he can be classified as a business user or a home computer enthusiast. And maybe he has interests in both areas.

So, a better solution might be to have node 1 as a small parallel sub-system, asking some probing questions which allow the user to be classified as business or non-business, or something between the two. Node 1 could then have three outcomes, business, home, and mixed use; each outcome leading to a separate node.

As you can well appreciate, any expert system using a node

Fig.2. A Node-based Expert System Shell

```
10 REMark Multi-node Expert System
20 REMark    by David Nowotnik
30 REMark    March, 1987
40 REMark
50 REMark    Initialise
60 RESTORE
70 READ t$: REMark title of system
80 READ nodes: REMark Number of nodes
90 READ q: REMark Number of questions
100 READ r: REMark Number of replies
130 DIM q$(q,40): DIM a$(q,2)
140 DIM o$(r,40)
150 LET nd=1: REMark Start at node 1
160 CLS : PRINT
200 PRINT t$: PRINT: GO SUB 4500
210 FOR i=1 TO q
220 READ x$: LET q$(i)=x$
230 LET a$(i,1)="n": LET a$(i,2)=" "
240 NEXT i
250 FOR i=1 TO r
260 READ x$: LET o$(i)=x$
270 NEXT i
1000 REMark Main Routine
1010 RESTORE (5000+500*nd)
1020 READ type: REMark Serial or parallel type
1030 IF type=1 THEN GO SUB 2000
1040 IF type=2 THEN GO SUB 3000
1050 IF p$="n" THEN nd=p: GO TO 1010
1060 PRINT
1070 PRINT "I recommend: ";o$(p)
1080 STOP
1090 REMark
2000 REMark Serial Node
2010 READ qu: REMark Number of branches
2020 DIM t(qu,3): REMark Question, yes/no data
2030 DIM f$(qu,2): REMark Where to go next on yes/no
2040 FOR i=1 TO qu
2050 READ x: LET t(i,1)=x
2060 READ w$: LET f$(i)=w$
2070 READ x: LET t(i,2)=x
2080 READ x: LET t(i,3)=x
2090 NEXT i
2100 LET item=1
2110 LET p$=q$(t(item,1))
2120 IF a$(t(item,1),1)="n" THEN GO TO 2160
2130 LET z$=a$(t(item,1),2)
2140 IF z$="y" THEN GO TO 2270
2150 IF z$="n" THEN GO TO 2230
2160 GO SUB 4000
2170 INPUT z$: IF z$="" THEN GO TO 2170
2180 IF z$(1)="y" OR z$(1)="Y" OR z$(1)="n" OR z$(1)="N" THEN
GO TO 2200
2190 GO TO 2170
2200 IF z$(1)="Y" OR z$(1)="y" THEN GO TO 2250
2210 PRINT "no": PRINT
2220 LET a$(t(item,1))="1n"
2230 IF f$(item,2)="q" THEN LET item=t(item,3): GO TO 2110
2240 LET p$=f$(item,2): LET p=t(item,3): RETurn
2250 PRINT "yes": PRINT
2260 LET a$(t(item,1))="1y"
2270 IF f$(item,1)="q" THEN LET item=t(item,2): GO TO 2110
2280 LET p$=f$(item,1): LET p=t(item,2): RETurn
2290 REMark This routine returns two variables
2300 REMark 'p' and 'p$'. 'p$' is 'n' for next
2310 REMark node or 'o' for outcome, and 'p'
2320 REMark gives the next node/outcome number
2330 REMark End of Serial Subroutine
2340 REMark
3000 REMark Parallel node
3010 READ qu: REMark Number of questions
3020 DIM t(qu)
3030 FOR i=1 TO qu
3040 READ x: LET t(i)=x
3050 NEXT i
3060 READ n: REMark number of outcomes
3070 DIM f$(n,3)
3080 FOR i=1 TO n
3090 READ w$: LET f$(i)=w$
3100 NEXT i
3110 DIM r$(n,qu): REMark Rules
3120 FOR i=1 TO n
3130 READ w$: r$(i)=w$
3140 NEXT i
3150 LET v$=""
3160 PRINT
3170 FOR i=1 TO qu
3180 LET p$=q$(t(i))
3190 IF a$(t(i))="yy" THEN LET z$="y": GO TO 3230
3200 IF a$(t(i))="yn" THEN LET z$="n": GO TO 3240
3210 GO SUB 4000
3220 INPUT z$: IF z$="" THEN GO TO 3220
3230 IF z$(1)="Y" OR z$(1)="y" THEN LET v$=v$&"1": LET
a$(t(i))="yy": PRINT "yes": GO TO 3260
3240 IF z$(1)="n" OR z$(1)="N" THEN LET v$=v$&"0": LET
a$(t(i))="yn": PRINT "no": GO TO 3260
3250 GO TO 3220
3260 NEXT i
3270 REMark Find a perfect match
3280 LET p=0
3290 FOR i=1 TO n
3300 IF v$=r$(i) THEN LET p=i
3310 NEXT i
3320 IF p=0 THEN GO TO 3380
3330 LET p$=f$(p,1): LET w$=f$(p)
3340 LET w$=w$(2 TO): LET p= VAL w$
3350 RETurn
3360 REMark Perfect match not found
3370 REMark Try to get best fit
3380 LET max=0: LET larg=0
3390 DIM k(n)
3400 FOR i=1 TO n
3410 LET k(i)=0
3420 FOR j=1 TO qu
3430 IF v$(j)="1" AND r$(i,j)="1" THEN k(i)=k(i)+1
3440 IF v$(j)="1" AND r$(i,j)="0" THEN k(i)=k(i)-1
3450 NEXT j
3460 IF k(i)>max THEN LET max=k(i): LET larg=i
3470 NEXT i
3480 IF larg<1 THEN PRINT "No match found": STOP
3490 LET p$=f$(larg,1): LET w$=f$(larg)
3500 LET w$=w$(2 TO): LET p= VAL w$
3510 RETurn
3520 REMark Same variables returned as serial subroutine
3530 REMark End of parallel subroutine
3540 REMark
4000 REMark Neat printing routine
4010 LET ii=0: LET LE= LEN (p$): LET LI=0
4020 LET x$=""
4030 LET ii=ii+1: IF ii>LE THEN  GO TO 4060
4040 IF p$(ii)=" " THEN  GO TO 4060
4050 LET x$=x$&p$(ii): GO TO 4030
4060 IF LI+ LEN (x$)>31 THEN  PRINT : LET LI=0
4070 PRINT x$;: LET LI=LI+ LEN (x$)
4080 IF LI<31 THEN  PRINT " ";: LET LI=LI+1
4090 IF ii<LE THEN  GO TO 4020
4100 PRINT : RETurn
4500 REMark Opening remarks
4520 PRINT: PRINT
4530 PRINT "I will ask you a series of"
4540 PRINT "questions. Answer with y"
4550 PRINT "for yes or n for no for"
4560 PRINT "each one until I provide"
4570 PRINT "you with the solution."
4580 PRINT : PRINT
4590 RETurn
4600 REMark
```

EXPERT SYSTEMS

approach for one particular application can adopt more than one strategy for solving the problem; it depends how the knowledge engineer wants to tackle the problem. Added to this, the framework for node-based systems must be flexible to link nodes in any desired way. The program in Fig. 2 has this capability; you'll see this once you understand how it works.

Table 1 gives the main variables in this node-based expert system. It also shows you how the DATA statements are arranged. By using this program as a frame, you can create your own expert system, by adding the knowledge base, and node logic; but you have to be careful to get the DATA statements just right. To allow you to test this expert system 'shell', and to demonstrate how it works, a small knowledge base is shown in Fig. 3. So, first type in the listing in Fig. 2. It will work with either Spectrum or QL, but omit the VAL functions (lines 3340 and 3500), and change the string joining '+' to '&' (line 4050) on the QL. Also, on the QL, the screen display will be neater without extra 'yes' and 'no' user responses printed on the screen (lines 2210, 2250, 3230, and 3240).

Once SAVEd, this will be your 'shell'. With this you can add all sorts of knowledge bases; like this one in Fig. 3.

Fig.3. Knowledge Base for Transport Example of the Node-based Expert system

```
5000 DATA "DETERMINE THE TYPE OF TRANSPORT"
5010 DATA 3,10,13
5020 REMark
5030 REMark Questions
5040 DATA "Does it travel on land"
5050 DATA "Does it travel on rails"
5060 DATA "Does it have just two wheels"
5070 DATA "Is it powered by an engine"
5080 DATA "Can it carry more than 6 passengers"
5090 DATA "Does it travel in the air"
5100 DATA "Does it only travel on the surface"
5110 DATA "Does it have fixed wings"
5120 DATA "Is it wind powered"
5130 DATA "Can it travel below the surface"
5140 REMark Answers
5150 DATA "train"
5160 DATA "motor bike"
5170 DATA "bicycle"
5180 DATA "coach"
5190 DATA "car"
5200 DATA "aeroplane"
5210 DATA "helicopter"
5220 DATA "glider"
5230 DATA "balloon"
5240 DATA "yacht"
5250 DATA "ship"
5260 DATA "submarine"
5270 DATA "hovercraft"
5280 REMark
5500 REMark Data for node 1 - Land, sea, or air?
5510 DATA 1,2
5520 DATA 1,"nq",2,2
5530 DATA 6,"nn",3,4
5540 REMark
6000 REMark Data for node 2 - Land Travel
6010 DATA 2,5,2,3,4,5,7
6020 DATA 6,"o01","o02","o03","o04","o05","o13"
6030 REMark Rules
6040 DATA "10111","01101","01001"
6050 DATA "00111","00101","00110"
6060 REMark
6500 REMark Data for node 3 - Air Travel
6510 DATA 2,4,4,5,8,9
6520 DATA 4,"o06","o07","o08","o09"
6530 REMark Rules
6540 DATA "1110","1100"
6550 DATA "0011","0001"
6560 REMark
7000 REMark Data for node 4 - Sea Travel
7010 DATA 2,4,4,7,9,10
7020 DATA 4,"o10","o11","o12","o13"
7040 REMark Rules
7050 DATA "0110","1100"
7060 DATA "1001","1000"
```

Knowledge base

The knowledge base in Fig. 3 creates a four node expert system, to determine mode of transport. A repeat of the example in part 2, you might think. Well, yes it is, but you should be able to see how this expert system reduces the number of questions (compared with the parallel system) but asks more relevant questions. The knowledge base is by no means comprehensive; as a learning example, I didn't think you'd want to type in pages of DATA lines. But type the knowledge base of Fig. 3 into your computer (QL or Spectrum), save it, then merge it with the shell from Fig. 2. When you RUN it, with the following analysis, you should see how it works.

With the question and response strings in q$() and o$() arrays, an extra array a$() is set up (line 130). This array holds an indicator to each question. If the question has not been asked, then the indicator reads 'n'; once a question is asked then the indicator reads 'y', and the response of 'yes' or 'no' stored alongside it. In node expert systems, it is quite possible for the same user answer to be required more than once. To avoid asking the same question twice, every time a question is selected, the indicator is checked (lines 2120 and 3190/3200); if it shows that the question has already been asked, then the previous response is read from the a$() array, rather than ask the user again. Every user response is recorded in array a$() for possible future reference.

The basic framework of the two main subroutines, starting at line 2000 for a serial node, and 3000 for parallel node, will be familiar to those of you who have followed this series, so I won't explain it in detail. Compared with the examples in part 2, much more information is stored in arrays. The array t() stores the question numbers relevant to that node; array f$() in the serial routine stores the information on where to go next (another question, another node, or to an outcome) for each yes/no response, the question/node/outcome number being stored in t(). In the parallel routine, f$() stores the outcomes, these being another node or outcome (response), and the number of that node or outcome. Array r$() in the parallel subroutine stores the rules for that node.

To try to understand the way it fits together, work through the node DATA lines in Fig. 3, and the order of data given in table 1. Do this while the transport example is running and this should clarify the logic of the program.

The program will always start on node 1 (line 150); thereafter, the next node (or eventual response) is determined as the outcome of the previous node. Once a new node is selected (returned as variable 'p' from either serial or parallel subroutine), the RESTORE command in line 1010 set up the data pointer to pick up the data for the new node. The first DATA item is always the node type, which in lines 1030 and 1040 sends the program off to the appropriate (serial or parallel) subroutine for that node.

Probability

Now for another way of trying to restrict questions to those relevant to the problem. Tagged on to the end of part 4 last month was a little piece of probability theory. I'll now give you a simple example of how this might be used to achieve a more logical selection of questions.

If you analyse the way you solve many problems, you may find yourself following this pattern. First of all you get some basic answers to give you a feel for the problem; in the transport example, determining whether the item of interest travelled on land, sea, or in the air (node 1) served that purpose.

Now, say the conclusion to this part of the exercise was air travel. Now as an expert, you know that the most frequently used form of air transport (of those known to you) is the aeroplane. So, you could assume that the answer will be an aeroplane, and ask questions which will confirm this. At each response to your questions, if the answer is as you suspected, then the probability of the outcome being an aeroplane is increased, and probability of it being one of the others decreased. Eventually, the evidence for aeroplane is so strong that it is pointless asking any more, and you (the expert)

give your answer.

Of course, if the answer to your questions points you away from your original theory, then you have to pick the next most likely candidate. That will be chosen on the basis of the original ranking of aircraft, and the user responses. Thereafter, your new line of questioning will be aimed at proving that your new target is the answer which is required.

To build this approach into an expert system, you, as the expert and knowledge engineer, have to provide the initial probabilities as new information is received. The example in Fig. 4 is a simple way of doing this.

As before, the listing in Fig. 4 will work on both Spectrum and QL. On the QL, the '+' in line 1050 should be '&', and line 560 can be omitted to give a more tidy screen display. When you run this program, arrays q$(), o$() are set up to hold the question and outcome strings. The initial probabilities of the four outcomes go into array p (). The data for this array is held in line 5110, and reflects the relative probability of seeing an aeroplane, glider, helicopter, or balloon in the air above my house. You may want to change these to reflect your regional bias. Array s() holds the factors by which each probability is altered should a 'yes' or 'no' answer be given for any question. With the deliberately small system chosen, this knowledge base is also quite small, but you can imagine how the complexity of this could increase with size of system. Array w() is a simple indicator to show whether a question has or has not been asked previously.

The important elements of the system are the initial probabilities, and the factors by which they are varied. In the example in Fig. 4, these factors are largely intuitive. For real expert systems, a lot of experience is necessary to get them right. As the new probability is the product of the old probability and the factor, these factors can have quite dramatic effects on the outcome, so use careful judgement in choosing these values if you decide to use the listing in Fig. 4 as a shell for your own system.

EXPERT SYSTEMS

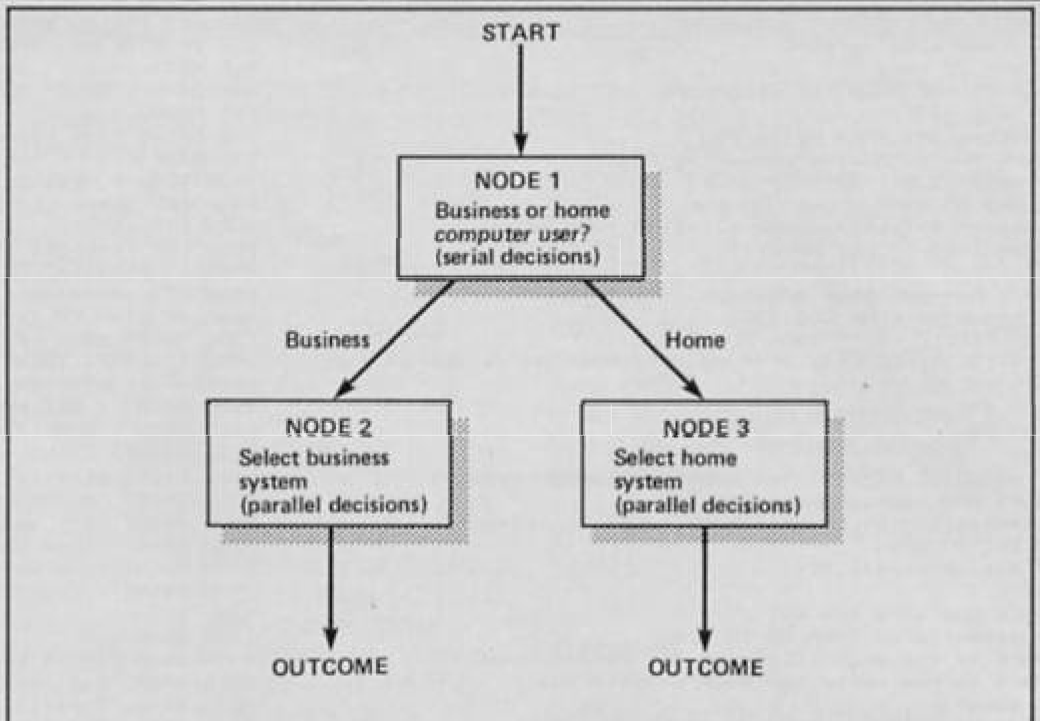


Figure 1. A simple example of a Node-based Expert System

The listing in Fig. 4 is reasonably well littered with REMark lines to explain what is going on; it follows the outline strategy given above. To decide on whether to go any further, the program compares the probability of the favoured item with the others. If the ratio is greater than 5 (line 750), then it can be assumed that the right answer has been found, and this is printed on the screen.

There is a certain danger in this approach. Ask a selection of biased questions (as this system does), and if that bias can favour one item enough to give a false answer. It could ignore a question which would rule out the favoured item, or create another favourite. So, again, care is needed with the values given to factors.

Goal

You may not have noticed but, in addition to the use of probabilities, this system differs in a major way from ones which I have presented previously. The earlier systems have been 'data-driven'. In other words, questions were asked without too much 'thought' (at least, by the computer) as to the outcome being approached. Both the serial and parallel approach ask questions in a pre-defined way. With the serial approach, each response is analysed as it is given; using the parallel system all responses are analysed at once. Either way, the expert system doesn't really know where it is heading until it gets there!

The listing in Fig. 4 is the first example in the series of a 'goal-driven' expert system. Here, the system finds (or is told) the most likely outcome, and selects the questions which can prove (or disprove) this theory. If it is proven to its satisfaction, then it need go no further, but if the theory is disproved, then it seeks the next most likely outcome, and biases its questions towards this new theory.

The very simple approach adopted by the example in Fig. 4 can be prone to error. So, in the final part of this series next month, I shall be taking a further look at goal-driven approaches, aiming to remove the possible misleading bits of the system presented this month.

Fig.4. Selecting questions using probabilities

```
10 REMark   Using probabilities
20 REMark   in an expert system
30 REMark     A simple example
40 REMark    by David Nowotnik
50 REMark       March, 1987
60 REMark
100 REMark   Initialise
110 LET qu=4: LET answ=4
120 DIM q$(qu,36): REMark Questions
130 DIM o$(answ,10): REMark Outcomes
140 DIM p(answ): REMark Probabilities
150 DIM s(answ,qu*2): REMark Store of Probabilities
160 DIM w(qu): REMark Indicator for questions asked
180 RESTORE
190 FOR i=1 TO qu
200 READ x$: LET q$(i)=x$: LET w(i)=0
210 NEXT i
220 FOR i=1 TO answ
230 READ x$: LET o$(i)=x$
240 NEXT i
250 FOR i=1 TO answ
260 READ x: LET p(i)=x
270 NEXT i
280 FOR i=1 TO answ
300 FOR j=1 TO qu*2
310 READ x: LET s(i,j)=x
320 NEXT j
330 NEXT i
340 REMark Main Routine
350 CLS: PRINT: PRINT "Modes of Air Transport"
```

```
360 GO SUB 1110
400 REMark Main loop
410 REMark Identify most probable item
420 LET item=0: LET prob=0
430 FOR i=1 TO answ
440 IF p(i)>prob THEN LET prob=p(i): LET item=i
450 NEXT i
455 IF p(item)>=1 THEN GO TO 900
460 REMark Now find most revealing Question
470 LET max=-1: LET pp=0
480 FOR i=0 TO qu-1
490 IF s(item,1+2*i)>max AND w(i+1)=0 THEN LET
max=s(item,1+2*i): LET pp=i+1
500 NEXT i
510 REMark Now ask that question
520 LET p$=q$(pp): GO SUB 1000
530 INPUT z$: IF z$="" THEN GO TO 530
540 IF z$(1)="Y" OR z$(1)="y" OR z$(1)="N" OR z$(1)="n" THEN
GO TO 560
550 GO TO 530
560 PRINT z$
570 PRINT
580 LET w(pp)=1: REMark Stop question being repeated
590 REMark Now update probabilities
600 LET f=(z$(1)="h" OR z$(1)="N"): LET tt=f+2*pp-1
610 FOR i=1 TO answ
620 LET p(i)=p(i)*s(i,tt)
630 NEXT i
640 REMark How sure are we?
650 IF p(item)<prob THEN GO TO 760
660 REMark If the probability of the main suspect
670 REMark is now worse, go back and try again
680 LET max=0
690 REMark How close is the nearest rival?
700 FOR i=1 TO answ
710 IF item=i THEN GO TO 740
720 LET sc=p(i)/p(item)
730 IF sc>max THEN LET max=sc
740 NEXT i
750 IF max<.2 THEN GO TO 900
760 LET ind=0
770 FOR i=1 TO qu
780 IF w(i)=0 THEN LET ind=1
790 NEXT i
800 IF ind=1 THEN GO TO 400
810 REMark No questions left
820 REMark So, confirm the most probable outcome
830 LET item=0: LET max=0
840 FOR i=1 TO answ
850 IF p(i)>max THEN LET max=p(i): LET item=i
860 NEXT i
900 REMark Print Outcome
910 PRINT
920 PRINT "The item is: ";o$(item)
930 STOP
1000 REMark Neat printing routine
1010 LET ii=0: LET LE= LEN (p$): LET LI=0
1020 LET x$=""
1030 LET ii=ii+1: IF ii>LE THEN  GO TO 1060
1040 IF p$(ii)=" " THEN  GO TO 1060
1050 LET x$=x$+p$(ii): GO TO 1030
1060 IF LI+ LEN (x$)>31 THEN  PRINT : LET LI=0
1070 PRINT x$;: LET LI=LI+ LEN (x$)
1080 IF LI<31 THEN  PRINT " ";: LET LI=LI+1
1090 IF ii<LE THEN  GO TO 1020
1100 PRINT : RETurn
1110 REMark Opening remarks
1120 PRINT: PRINT
1130 PRINT "I will ask you a series of"
1140 PRINT "questions. Answer with y"
1150 PRINT "for yes or n for no for"
1160 PRINT "each one until I provide"
1170 PRINT "you with the solution."
1180 PRINT : PRINT
1190 RETurn
1200 REMark
5000 DATA "Is it powered by an engine"
5010 DATA "Can it carry more than 6 passengers"
5020 DATA "Does it have fixed wings"
5030 DATA "Is it wind powered"
5040 DATA "aeroplane"
5050 DATA "helicopter"
5060 DATA "glider"
5070 DATA "balloon"
5100 REMark Initial Probabilities
5110 DATA .35,.25,.25,.15
5120 REMark Factors to change probabilities
5130 REMark on yes/no answers
5140 DATA 1.5,0,1.5,.5,1.5,0,0,1: REMark Aeroplane
5150 DATA 2,0,1.2,.8,0,2,0,1.2: REMark Helicopter
5160 DATA 0,2,0,2,1.5,0,1.5,0: REMark Glider
5170 DATA 0,3,.8,1.2,0,3,2,.2: REMark Balloon
```

CROSSFIRE

On tape

May I first of all congratulate you on the continuing quality of ZX Computing Monthly. There is always much of interest in each issue. The features, articles and reviews are generally well written and presented. (Which is more than can be said of many of your competitors). I have learnt much from the programming features which are of great use in building a reference library of usefull utilities.

I have but one complaint, because of the nature of many of the programming series and utilities the listings are often lengthy which leads to errors when entering them and there simply is not enough time to type them in before the next issue, a recent example being the Streams and Channels series. These would be excellent candidates for transfer to tape. In particular I would like to see Specword and the Technical Graphics series in tape form.

The April Fools article was well done. Despite the error in the listing! I am pleased that you are supporting the 128k machines and have not abandoned the earlier Sinclair computers and the QL. (Although where was Page 81 in the April issue?) I still marvel, occasionally, at what can be done with the ZX81.
P. R. King,
Wells.

Thank you for your suggestion Mr King and thanks to all the ZX readers who wrote to us nominating listings for our tape service.

Unfortunately the march of time has caught up with the ZX81 and quality material for the ZX81 is getting very hard to find. So the 81 page will appear no more but look out for items on the news pages if new products appear and of course 81 owners can still make their views known via Crossfire and get their problems solved via Crosswires.

Short Cuts

Well done! You are quite right in awarding 'Star Cut' to Chris Richards in the April issue. I have now at last been able to use handy Tasword 2 again on my Spectrum 128/Opus Drive to print out in 48K mode to a serial printer at 1200 baud using the 128's RS232 Port.

For the record I also converted the 64 column fonts as in the March issue (Better Letter), and use an RGB Monitor via the Spectrum 128 RGB socket thus giving a very complete package.
D. C. Jenkin,
Devon.

Favourite Utility

First of all I would like to congratulate you on your interesting Spectrum journal.

In answer to Alan davis's question (ZX February) on what utilities readers have found most useful over the years I would like to say that my personal favourite is Keydefine by Scientific Software. I have been using it for a very long time and made many additions. It consists of a lot of functions such as such as trace renumber, delete lines, code to data lines etc at the stroke of 2 keys. The utility is in machine code, but personal functions may be added via a basic program. I think the author, merits admiration for this ingenious utility.
U. Lenoir,
Baden Baden,
W. Germany.

BUILD A BETA BASIC

John Wase reviews the upgraded Beta Basic version 4.0

Beta Basic 4.0
Betasoft
£15.95

To say the least, it's a headache to write a review of this program, because, like Tasword, it has a long and noble line of predecessors; nevertheless, some readers clearly don't use it, or haven't even come across it. So I'll start with a short account of Beta Basic 3.0 for the 48K Spectrum (or the 128 or +2 in 48K mode) and then discuss some of the extra 128K program features.

Effectively, what Beta Basic does is to enhance Spectrum BASIC by adding many new commands and functions. There are so many I cannot give very much detail but merely select from some seventy extra commands, most multiple, and twenty-six extra functions which appear as still further keywords. The essential Sinclair format remains the same, although much has been added. For instance when the computer stops after executing a command, type in the figure nought, and up comes the keyword EDIT on the input line (no keyword or line number starts with zero). Add the line number, ENTER, and there it is, waiting on the bottom lines to be edited. Saves all this listing to a line, breaking, and so on. In addition, the cursor is improved, the editing cursor moves up and down as it ought to, CSIZE changes the size of the lettering on-screen, you can arrange either to type in keywords, individual letters, or even a mixture of modes, and JOIN and SPLIT program lines ... But I digress. The most important bits for any BASIC programmer are things like the enhanced loops (DO. . .LOOP: DO WHILE condition; DO UNTIL condition; EXIT IF). Structured programming is also enhanced with commands like IF. . . THEN. . . ELSE, ON (to select a line number or a statement from a list) and LIST FORMAT which provides automatically indented listings to emphasise structure. But undoubtedly the most elegant and useful facility is the very full implementation of procedures, (allowing really well structured programs) and including the keywords PROC, DEF PROC, END PROC, with LOCAL, REF parameter, DEFAULT variable value and ITEM function. Global and LOCAL variables can be used, and parameters can be passed by reference; even arrays, so that you need hold only one copy in memory.

There is also an assortment of graphics commands (CSIZE; GET a screen area, POKE it into memory, ALTER various attributes; DRAW TO; FILL, ROLL and SCROLL any parts of the screen, and WINDOW). Beta Basic's improved SAVE, LOAD, MERGE and VERIFY all permit use of parts of programs (SAVE a library of procedures) and allow simplified syntax for microdrives or discs (LOAD 1; "program", or use DEFAULT).

You can SAVE, MOVE, or COPY part of an array to another array, or ALTER its size without data loss. DELETE deletes part of an array and SORT gives very fast alphabetic or numeric sorting. INARRAY and INSTRING functions search arrays and strings whilst LENGTH gives an array's dimensions and location to check if it will fit in memory.

Finally a comprehensive set of toolkit commands includes block move, copy or delete, a comprehensive RENUMber command and ALTER, a search and change command. Those of you who don't know Beta Basic and who actually write BASIC programs (useful if you want to understand what the program's about when you look at it next month) will be wondering "What's the catch?" The answer is all this lot takes quite a chunk of memory in the 48K Spectrum, although it still leaves about 22K of memory for your BASIC program. (How many BASIC programs are longer than this?). The crunch comes when you want to handle large amounts of data, particularly large arrays. Which is where I came in: with Beta Basic 4.0.

Beta 4.0

The last but one Beta Basic Newsletter (contact Betasoft for details of this service) was rather jaundiced about the 128 or +2's very restricted memory paging, but has now relented, mainly because the data handling restrictions I just mentioned, and the ability to handle large arrays are among the features of the new program which I tested.

The presentation, a pleasant surprise, is still 48K-type with an input line at the bottom. The full-screen editor of the 128K mode is a bind, because it has to redraw the whole screen for each command and invariably loses the first few letters of the next command typed in as there is no input buffer. Beta Basic 4.0, like previous versions, allows keywords to be typed in, in full, so with its improved editing facilities, its display is probably better.

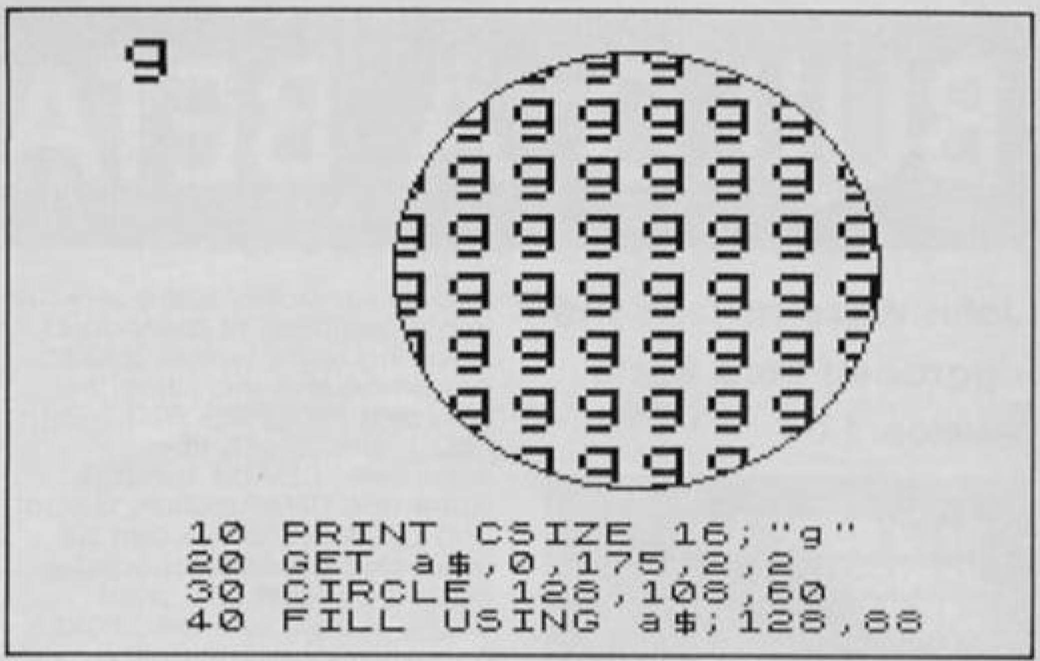
Arrays

Let's start with the serious stuff; arrays. Although the display looks 48K, the new keywords are distinctly 128Kish. So **DIM! fred!, (1000, 60)** creates a 60K RAM disc file. You can then sort it using SORT !, (e.g. **SORT ! fred$()**). **LET !** enables you to assign a number or a string to a specific element in the array (e.g. **LET ! fred$ (20) = "hello"**) and PRINT INARRAY ("fred$", "test") found "test" in fred$ in about 0.66 sec/1000 strings if all length (60) was searched, and about 0.26 sec/1000 strings if position in the strings was given. LIST ! AND LLIST ! list arrays to screen or printer (e.g. LIST ! "fred$"), LIST ! being the most important feature of these new array-handling facilities as it provides the ability to LIST ! to any peripheral with direct access to streams/channels.

In effect, this means you can set up files in RAM which are essentially random access, manipulate them incredibly rapidly whilst still in RAM and dump them out again to storage. Even microdrive owners can have random access facilities now, and at a speed many a PC owner would envy. The final extra RAM disc command is INPUT ! to LOAD a file back from a device to RAM disc (e.g. **OPEN # 5;"m";1;"file-name": INPUT # 5;!"fred$")** which LOADs fred$ back to the RAM disc array. The complete collection of RAM disc commands is SAVE !, LOAD !, MERGE ! and ERASE ! (same as in Spectrum 128 BASIC): CAT ! (now gives free disc space; normal, 78K); INPUT ! and LIST ! to move stuff to and from bulk storage. DIM ! to set up a RAM disc array, INARRAY to sort RAM or RAM disc array and SORT ! to sort a RAM disc file.

Beep!

In addition, there are some fun bits. A fast CIRCLE (about twelve times faster than Sinclair's) and DRAW (about 2.5 times faster: it will also draw on the bottom two lines of the screen which the new PLOT can also use); a new FORMAT to allow a ZX printer to work correctly in 128K mode; FILL USING gives fast patterned, even stippled fills. I filled a quick circle with a nice redbrick pattern. Whilst Beta Basic of course supports the new PLAY command which is great for music, but which stops the computer every time it sounds. It also has a totally new, super-fun command; an interrupt-driven BEEP ! which can have up to six specifiers.



A screen produced with some of the new Beta commands.

There isn't room here to describe them in detail, but all the facilities of the sound chip are at your disposal. The big advantage over PLAY is that BEEP ! is interrupt-driven: Beta Soft provided me with a few lines of Beta Basic illustrating several features rather nicely; a rapid circle, rapid filled and bounced up and down between two lines at ever-increasing velocity, raising the pitch of the "boing" at each bounce. With PLAY, the poor old computer and ball would have stopped at each surface to make the noise; with BEEP ! the ball carried on bouncing.

All in all, this is a super program. It provides an unbelievable number of facilities, and comes as a package containing the 48K Beta Basic 3.0 and the 128K Beta Basic 4.0 recorded together.

The complexity of these programs is indicated by the size of the instruction manuals; 88 closely printed A5 pages for the basic 48K version 3.0, and a supplementary manual containing a further 30 or so pages for issue 4.0, all tightly packed with information and carefully laid out with illustrative programs.

The whole comes packed in a stylish wallet from Beta Soft, 92 Oxford Road, Moseley, Birmingham, B13 9SQ. Write to them for details of disc versions and upgrades for existing owners.

Beta Basic 3.0 was superb, one of the classic programs for the 48K Spectrum. Beta Basic 128 is even better; it's solid gold. Get it!



DISCIPLE DATA

Machine Code Programming with Disciple — by John Wase.

Disciple's Command Codes which I mentioned recently are used to gain access to the disc controller subroutines, enabling serious programmers to drive the disc directly from machine code. This rather ponderous start in fact begs a couple of questions which I'm frequently asked:

1) Are the Command Codes different from the hook codes I've heard about?

2) Do Sinclair's hook codes work on the Disciple?

The answer to the first question is "Yes". They're different, they're simpler to use and they're exclusive to Disciple. This is why they're called Command Codes — to distinguish them from Sinclair's hook codes, which, of course, are designed to work on Interface 1. Nevertheless Sinclair's hook codes will almost all work, too (question 2). Here, therefore, is a table of Command Codes (figure 1). These apply both to version 2 and to the new version 3. Do note that all Command Codes corrupt register HL and that if an error is found — either when using a Command Code or a Sinclair hook code, then a return is made with the carry flag set and the register "A" holding the error number for interpretation by the calling routine.

Utilities

If you have a Disciple and use **Beta Basic**, contact Betasoft, who will be able to supply the POKEs to enable your version to run. I understand that when the ROM has been finalised there is the possibility that Betasoft will either issue a code patch or a special version for Disciple users.

I understand that Saga's **The Last Word** is being converted for the Disciple. As soon as I can get hold of a review copy, I'll give you more information.

Queries

I have here a supply of general queries which I have gleaned from the manufacturer's files. The enquiriers will of course remain anonymous, though if you write to this column publishing your name is fair game unless you specifically forbid it — you have been warned! More to the point is the fact that these queries represent general problems that readers have had, and the information could, perhaps, save you a lot of difficulty. The principal preoccupation of users is, as always, with connecting up Spectrum, printer and word processor.

In the column last month I mentioned Tasword. Tasword 2 is still a favourite processor and can be transferred to Disciple fairly readily. Tasword 3 gives difficulties because of the way it interacts with Interface 1 ROM. There is no firm news from Tasman yet of a conversion. Tasword 128/Tasword +2 will, I suspect, be usable with Disciple Version 3 using the 128K snapshot facility — we are investigating. . .

So, back to Tasword 2. The most important thing to remember is that to use the printer control codes, you need to put in the appropriate POKE @ 6 commands. When you POKE @ 6,1 the output from the printer port is unaltered; all the absolute values of the ASCII codes fed to it are output again unaltered. POKE @ 6,0 tells the printer driver to look for attribute commands such as TAB and AT, and to tokenise keywords. It is thus essential for Tasword 2 to incorporate POKE @ 6,1 somewhere in the program before you use the printer.

One enquirer had combined it with a CHR$ 27 instruction, which is, of course, wrong. However, he also submitted a short piece of program which gives Tasword 2 the facility to provide multiple copies and also left hand margin control. Here it is (figure 2). Finally, an enquirer mentions that Tasword 2 asks for input of interface control codes. This is unnecessary if you use the POKE @ statement (Line numbers 279 and 281, Figure 2).

Interfaces

Talking of interfaces, another enquiry came from someone who wanted to use the Disciple with the Euroelectronics' ZX LPRINT III printer interface. Regrettably this is not possible, for on power-up, both the Disciple and the interface are tryint to access the same ports on the computer. This locks the whole thing up and prevents even the system disc loading.

That reminds me of one or two bugs that have cropped up in Disciple version 2. VERIFY can give the error message "sector data lost" and the file is then unusable. There are really two bugs here: there's a bug in the VERIFY command and some of the error statements appear out of order. Both these problems have been fixed in issue 3. Meanwhile, just take care to save everything twice — on two separate discs!

Another problem concerns erasing a file by character string. **ERASE D* N$** does not work, nor, for that matter, **LOAD D1 N$**. The manual omits a separator (;) before N$ — insert one and all will be well. COPY D1 "filename" works for small files, but if a second "Change Disc" message appears, the utility tries to write on the source

Fig. 2 Changes in Tasword 2 for multiple copies and L/H margin control.

```
Changes in Tasword 2 (Disciple version) for multiple copies and
left hand margin control

 251 LET i=14:PRINT AT i,0;"No of copies (1)": GOSUB 6000: IF a$
= ""THEN LET a$="1"
 252 LET k=VAL a$
 253 LET i=16: PRINT AT i,0;"Left margin (";PEEK 60927;")": GO S
UB 6000: IF a$<>"" THEN POKE 60927,VAL a$
 260 CLS : PRINT AT 20,0;"Press the q key to quit printing"
 270 LET x=60049: GO SUB 950
 275 LET c=PEEK 62470: IF c<>0 THEN LPRINT CHR$ c
 278 FOR i=1 TO k
 279 POKE @6,0
 280 RANDOMIZE USR 60038
 281 POKE @6,0
 282 NEXT i
```

disc. This, again, is a bug which has now been corrected in version 3, as have the bugs in LPRINT and LLIST. Details of version 3 upgrading should by now have appeared in the Users' Club Newsletter; if you have not received one, make enquiries of Bob Brenchley (details last month).

Some readers have written in to say that their Disciples won't work. The symptom is that no disc commands (e.g. CAT 1) are accepted from the input line — the syntax check throw up a question mark. Before blaming Rockfort, see if your Spectrum will work in a friend's Interface 1 or Discovery. If not, suspect the M1 line of the Z80 CPU. A whole lot of chips with suspect M1 lines have been installed in Spectrums and all is fine until the M1 line is needed for paging in a further ROM. In addition, one or two reports have been received of issue 2 Spectrums in which the lead was missing from tag 27 on the Z80 CPU (the M1 line) to 24B on the edge connector. Resolder (not a job for the inexperienced amateur) and the problem goes away. Finally, before dashing to your friend, do make sure the edge connector is clean — a pensil eraser will often do the trick.

Fig. 1 Command codes for the Disciple.

HXFER	33h
DFSN	34h
HOFLE	35h
SBYT	36h
HSVBK	37h
CFSM	38h
PNTP	39h
COPS	3Ah
HGFLE	3Bh
LBYT	3Ch
HLDBK	3Dh
WSAD	3Eh
RSAD	3Fh
REST	40h
HERAZ	41h

HXFER — 33h
Transfer the file description and header to the Disk File Channel Area (DFCA). Register IX must point to the first byte (DSTR1) of the user's 24 byte file information area (UFIA). Expect this command code to use the registers HL, DE, BC and A.

OFSM — 34h
Open file sector map with the information contained in the DFCA. Eleven bytes from NSTR1 are used to set up the file name. A return is made with the disk buffer pointer (RPT) set to the start of the disc buffer RAM (DRAM).

HOFLE — 35h
Open a file. IX must point to the user's file information area (UFIA). This code combines the functions of HXFER and DFSM, also setting the 9 bytes of header information from HD00 to the beginning of the file. This header to the file allows the normal Load D1 " " syntax to call the file.

SBYT — 36h
Save the byte in register "A" to DRAM at the location pointed to by RPJ. If the disc buffer is full, an automatic sector save to disc will take place, and there will be an update of RPT and the value entered at the beginning of the DRAM.

HSVBK — 37h
Save a block of data to the disc when register DE points to the start of the data and register BC holds the byte count.

CFSM — 38h
Close file sector map. This routine empties the DRAM and copies the header area onto the directory, closes the file, then updates the directory.

PNTP — 39h
Output the 8-bit code in register "A" to the printer port, after checking busy status of the printer.

COPS — 3Ah
Copy the screen RAM to the printer in high resolution screen graphics as set up under the graphics control variables.

HGFLE — 3Bh
Get a file from disc. IX register must point to the start of the UFIA. The return is made with the first sector of the file loaded into DRAM and RPT pointing to the first byte (usually the header information).

LBYT — 3Ch
Load the byte pointed to by RPT from DRAM, and return with it in register "A" and RPT updated. If the buffer is empty, another sector is read from the disk. (Be careful: is there another sector?)

HLDBK — 3Dh
Load a block of data from disc to the memory pointed to by the register DE with the block count in register BC.

WSAD — 3Eh
Write sector at DE. Writes the contents of the DRAM at the track pointed to by D and the sector pointed to by E, and sets the pointer to the start of DRAM.

RSAD — 3Fh
Read sector at DE. Reads the sector at track D, sector E into the DRAM, and sets the pointer RPT to the start of DRAM.

REST — 40h
Restore disc drive to Track 00. The drive number is taken from DFCA.

HERAZ — 41h
Erase the file on disc with the information contained in UFIA — so make sure that IX points to this.

Figure 3

```
Main modifications to Tasword 2's microdrive
version for Disciple: This configuration is for use with a
serial printer connected to the RS 232 port on interface 1.

 279 OUT 31,0: FORMAT  "b";9600:
OPEN #3;"b"
(Note: insert your baud rate  if
different)
 281 CLOSE #3: OUT 31,16
 700  CLS  :  LET  a$="tasword":
SAVE d1;"Autorun" LINE 15
 780 VERIFY d1"Autorun": PRINT A
T 21,0;"tasword: basic O.K.;";AT
 19,0;
1000 LET b=FN p(62216): CLS : CA
T md!
1005 PRINT AT 20,1;"Name of text
 file for saving?": LET i=21: LE
T j0=10: GO SUB 6000
(Note: the 0 in the variable j0
is a zero)

2000 CLS : CAT md!: PRINT AT 19,
1;"Type the name of the text fil
e";AT 20,9;"and press ENTER"
2020 LET j0=10: LET i=21: GO SUB
 6000: IF a$ ="" THEN GO TO 2020
5000 CLS : CAT md!: LET jo=10: P
RINT AT 18,jo;"Current Drive=";m
d
(Note: the o in the variable jo
is a letter)
5010 PRINT AT 19,1;"Type new dri
ve # and ENTER, or";AT 20,1;"Pre
ss ENTER for current drive"
5020 LET i=21: GO SUB 6000: IF a
$="" THEN CLS : GO TO 25
6000 LET a$="": PRINT AT 21,J0;
FLASH 1;" "
```

Tasword

Last month I mentioned problems concerning the Disciple's incompatibility with microdrive versions of Tasword. Meanwhile, for those who are desperate, here is a temporary stopgap; conversion instructions for Tasword 2 by Brian Ashley of London. If you already have a customised microdrive version, connect interface 1, press Disciple's inhibit button once to enable the disable function then enter **OUT 31,0.** Load your own customised Tasword 2 from microdrive **(LOAD • "m";1; "run"** — note that "run" is typed in full). Enter BASIC through the main menu and modify the program using the lines in figure 3, and save; this will print files through

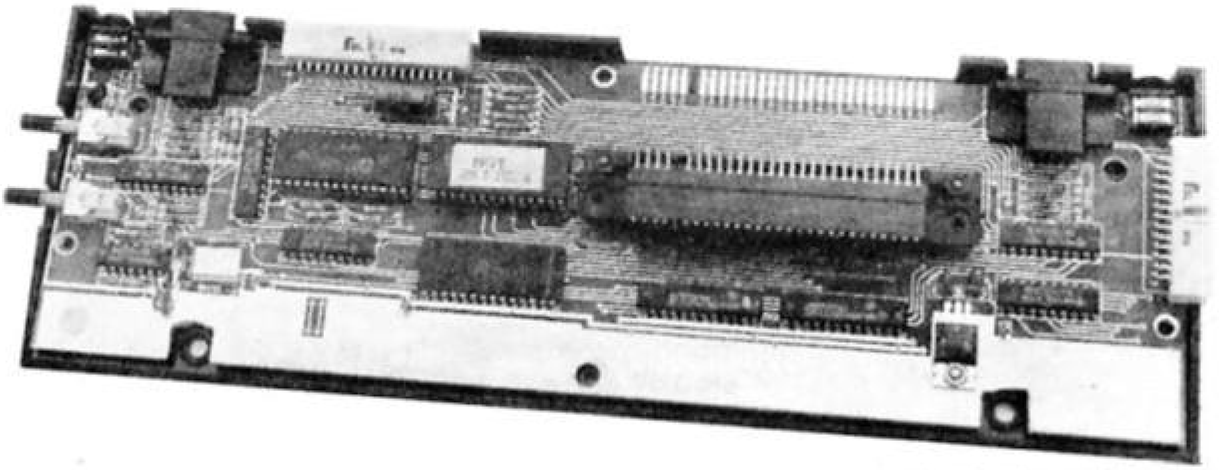


Figure 4

```
Program modifications to transfer existing files  from
microdrive to disc.

  26 OUT 31,0
1000 LET b=FN p(62216): CLS :OUT
 31,16: CAT md!
```

Figure 5

```
Modifications to use the Disciple's Centronics printer
port.

 279 POKE @ 6,1
 281 POKE @ 6,0
8200 LET d=1

When Tasword asks which  printer
interface you're  using,  select
Sinclair.
```

the original interface 1 RS232 port. Transferring existing files from microdrive to disc is done by further modifying the program according to figure 4: delete line 26 and the **OUT 31,16** in line 1000 when you've done. Disciple's parallel port can be used instead of the interface 1 RS232 by substituting the lines in figure 5 and selecting "Sinclair" when asked about interfaces. If the keyboard locks after printing a file, try turning the printer on first, then booting up the system. Brian's program has the 'O' and the 'j' different in different lines as marked; my version has them all lower case 'j' and 'zero'. So check your original before modifying.

Finally, if your version is on tape, load it, convert to microdrive syntax as detailed in the instruction book, make the further modification in figures 3 and 5 and the program is now configured to run on Disciple, using the centronics port provided.

Tailpiece

If you own a Disciple, you are invited to submit comments, programs or information: please sent it to the Editor, ZX Computing, marking your envelope "Disciple". That's all for this month, then. Be seeing you!

Midi Interface

Fast emerging as the home musician's supplier, Cheetah venture into a more sophisticated application.

MIDI interface
Cheetah
£49.95

Perhaps the most exciting development in the musical field has been the advent of keyboards and synthesizers which use chip based technology. As this is similar to that used in computers the possibility of connecting the two has existed for some time now.

In home computing terms the advent of the MIDI standard — Musical Instrument Digital Interface — has meant that for £100+ you could link your Spectrum to a keyboard equipped with MIDI sockets and perform a variety of control functions such as editing synth sounds or, more generally useful, set up musical sequences.

Up until now a sequencer program would either be *real time* (you play and record like using a tape recorder), or *step time* (where you enter notes one at a time at the keyboard).

Such programs, because of their specialist application and limited market, cost a lot. Usually between £20 and £50 each and were aimed at recording studios and music enthusiasts. However such MIDI keyboards have dropped dramatically in price and can be purchased from around £150+.

Cheetah have now identified a more general market for MIDI users and produced their own interface complete with sequencing software for £49.95, by far the cheapest available.

The Interface

This is the usual dead ended plastic case which plugs into the user port of the Spectrum. At the rear are three 5-pin DIN MIDI standard sockets providing IN, OUT and THROUGH connections. These are bolted on behind the plastic case and I found that some of my DIN plugs did not have enough reach to plug in. The holes in the plastic casing are not wide enough to allow the plastic covering of the DIN plug to pass. The result is connections which are not very secure. Provided the set up is stable then there is no problem, but if the computer moves then a lead may fall out. This even applies to the supplied DIN lead.

The through socket also doubles as a synchronisation socket for sequencers sending 96, 48 or 24 pulses per quarter note via pins 1 and 3.

Considering the price the unit is very good, although the socket holes should be enlarged.

Reading the specifications for the software aroused my curiousity. Presented in a minute, eyestraining 18 page 'instruction manual' the program seemed to offer a rather sophisticated range of operations.

"The system will allow you to record up to eight tracks of music, each track having 16 verses stored in it, each verse being from 1 to 64 bars in length. Each track may be assigned to any one of 16 music channels and a song constructed and played on any or all of the eight tracks simultaneously."

Not only that but the music can be entered in real time or step time and edited in step time regardless of which method of entering the music was used.

Each verse can contain a single note sequence or a polyphonic sequence, a very powerful choice, and there are ten degrees of quantisation or auto time correction, ranging from none to half note accuracy.

Options

On loading the program after connecting up the interface, the screen offers a microdrive copy to be made. Declining this option presents you with a full 22 lines of options including: Record a verse, Track and verse to record, Adjust a verse length, Link two, or copy one verse, Delete a verse, Reverse a verse and many more.

Selection of an option is by moving a bar up and down the screen and pressing Enter. This often leads to a sub-menu where parameters or further options can be selected also using the cursor keys.

In use

I spent a while with it using the Cheetah MK5 keyboard as an input source and a Casio CZ101 synth as the sound source connected to the MIDI out. The interface itself does not produce any sound.

It is impressive and I enjoyed working with it. Some respectable music was created fairly quickly, and every "I bet I can't do . . . " thought that occurred to me was proven wrong. I couldn't think of anything that it couldn't do!

There are however a couple of minor shortcomings. First, it does not automatically turn off all notes when you stop from the sequence play option and all too often the synth continued to play a note indefinitely after leaving that option. The only recourse was to turn off the synth. If you have several keyboards set up with specific control features then this could be rather irritating and time consuming.

Although the playing range is eleven octaves only the five from 4 to 8 actually operated with my synth, lower and higher notes being played at the highest octave it was capable of. I suspect within a few limits this will apply to many keyboards apart from the full-sized piano simulators.

When in step-time or edit mode the music is displayed in pianola fashion, bar lines which vary in length depending on how long the notes are. These are displayed one octave at a time and are positioned vertically. Bars are defined by double vertical lines split by single lines depending on the time signature. Horizontal and vertical editing cursors allow you to select the note you wish to modify.

Conclusion

For home and general use the interface is brilliant, well worth the money and it extends the use of any MIDI equipped instrument fantastically. To the enthusiastic amateur or for semi-pro applications it is excellent value.

Although the MIDI codes are standardised, unfortunately the MIDI interface ports which make the connection to the computer are not. So far every interface I've tried uses a different combination for Status, Send and receive, to my disappointment this interface is no exception. You cannot use XRI or EMR MIDI software with the Cheetah interface. A pity as it would have made the unit more viable.

It is compatible with the ZX Interface 1 and microdrives and, I believe the Discovery Disk drive.

All Cheetah have to do now is bring out a good independent sound generator on the lines of the TX7 or FB01 for under £100 and a complete studio set up will be available at a reasonable price.

Bonabanner & Utility 'A'

Ray Elder looks at two unusual programs from Chezron Software.

Bonabanner
Utility A.
Chezron Software
£3.50 each.

Chezron Software is a small mail order company who have produced some general purpose programs, mostly in Basic, for specific applications.

UTILITY 'A' is primarily designed to enhance microdrive handling by providing a "front end" menu system for incorporation into your own programs. This performs as per specs in that it is a ready built CAT and menu from itself when saved on any microdrive as file 'run'.

It is fairly versatile and will handle up to eight microdrives. Also supplied is a routine to add to your programs under development to auto save and update the cartridge.

Thrown in for good measure are two other programs, Time Count produces tables of tape counter equivalents for users of more than one cassette player and Autodraw is a pretty pointless diversion into kaleidoscope graphics.

All this is moderately useful, especially if you use microdrives and cannot be bothered to get to grips with them and write your own routines. At £3.50 it is worth it if you want something along these lines.

Bonabanner is a half way house between a full blown artist program and producing graphics from Basic. Written mostly in Basic it allows you to draw on a third of the screen using the block characters and UDG's and then it produces a Basic line of REM, PRINT or DATA with it in.

The drawing screen is easy to handle and works well, with practise (and as shown in the Demo) some quite impressive pictures can be produced. Two or three pictures can be combined to form a larger one.

Some possible uses are to produce loading pictures which appear faster and more impressively than a screen, decorate and enhance your own screens or even produce graphics for your own Basic adventures.

Each picture uses about 600 bytes and often, by using the supplied compressor, as few as 90 bytes.

Priced at £3.50 it is again good value if you want this sort of program, it does ease the task of producing graphics from Basic, but do you need it enough to part with your money. I have no doubt that many readers will find it useful.

Chezron Software: Contact Sandy at 605 Loughborough Rd. Birstall, Leicester LE4 4NL or phone Ron on 0533 661541.

MULTI-S

Alan Davis presents an all purpose scrolling utility.

Broadly speaking, I suppose there are really only two ways of presenting text on screen in an adventure game. Probably the simplest is to use straightforward PRINT statements in which strings of text are printed from some specified screen line downwards. This is a method more commonly encountered in older games, perhaps. There's nothing actually wrong with it as a system, except for the (relatively minor) disadvantage that the relevant part of the screen needs to be cleared completely before each fresh printing — obliterating whatever went before, of course.

The alternative method — and the one almost universally used these days in commercial games — is the continuous scrolling one. With this system, text is printed one line at a time to a particular screen position, and the screen is scrolled upwards by one line between each printing. In a text-only adventure this isn't difficult to achieve because the well-known USR call to the Spectrum ROM (3582) can be used to scroll the entire screen in between the printing of each line of text. However, suppose you're writing

Listing 1

Assembler routine.

```
*HISOFT GENS3M2 ASSEMBLER*
       ZX SPECTRUM

Copyright (C) HISOFT 1983.4
All rights reserved

Pass 1 errors: 00

        10 *C-
        20 *D+
        30 ;Routine for printing and scrolling text in any
        40 ;window, anywhere on the screen
64800   50         ORG  64800
        60 ;Initialisation routine
64800   70 INIT    LD   A,(TOP): Top of window
64803   80         LD   B,A
64804   90         LD   A,(BOT): Bottom of window
64807  100         SUB  B
64808  110         INC  A
64809  120         LD   (LINES),A: No. of lines to be scrolled
64812  130         LD   A,(LEFT): Left hand column of window
64815  140         LD   B,A
64816  150         LD   A,(RIGHT): Right hand column of window
64819  160         SUB  B
64820  170         INC  A
64821  180         LD   (CHRS),A: No. of characters per line
64824  190         RET
       200 ;Main scrolling routine
       210 SCROLL
64825  220         CALL INIT
64828  230         LD   HL,TABLE: Start of address table
64831  240         LD   BC,(TOP): Move the "pointer" to the
64835  250         LD   B,0: appropriate position in the table
64837  260         ADD  HL,BC: and store it in (POINT)
64838  270         ADD  HL,BC
64839  280         LD   (POINT),HL: Pointer position stored
64842  290 LOOP    LD   HL,(POINT)
64845  300         LD   E,(HL)
64846  310         INC  HL
64847  320         LD   D,(HL)
       330 ;Address of start of screen line now in DE
64848  340         PUSH HL
64849  350         LD   HL,(LEFT)
64852  360         LD   H,0
64854  370         ADD  HL,DE
64855  380         EX   DE,HL
       390 ;Address of left hand side of window now in DE
64856  400         POP  HL
64857  410         LD   A,(LINES): Decrease the line count
64860  420         DEC  A
64861  430         CP   0: Have we scrolled all the lines yet?
64863  440         JP   Z,BLANK: Quit this loop if we have
64866  450         LD   (LINES),A: Store the line count
64869  460         INC  HL: Move the pointer to the next item
64870  470         LD   (POINT),HL: in the table. Save position
64873  480         LD   C,(HL)
64874  490         INC  HL
64875  500         LD   B,(HL): Start of next line down in BC
64876  510         LD   HL,(LEFT)
64879  520         LD   H,0
64881  530         ADD  HL,BC
       540 ;HL now points to the screen address 8 pixels below
       550 ;the one held in DE
64882  560         LD   B,8: 8 pixel lines to be transferred
       570 ;Now move 8 pixel lines up the screen by 8 pixels
64884  580 TRANS   PUSH BC
64885  590         LD   BC,(CHRS)
64889  600         LD   B,0
64891  610         PUSH HL
64892  620         PUSH DE: Save all registers
64893  630         LDIR : Transfer the line of pixels
64895  640         POP  HL
64896  650         POP  DE
64897  660         LD   BC,256
64900  670         ADD  HL,BC
64901  680         EX   DE,HL
64902  690         ADD  HL,BC: Move HL and DE down one pixel
64903  700         POP  BC
64904  710         DJNZ TRANS
       720 ; One line of characters has now been transferred
64906  730         JP   LOOP: Back for next line of characters
       740 ;Scrolling finished. Now erase last character line
64909  750 BLANK   LD   B,8
64911  760 LOOP2   PUSH BC
64912  770         PUSH DE
64913  780         LD   A,(CHRS)
64916  790         LD   B,A
64917  800         XOR  A
64918  810 LOOP3   LD   (DE),A
64919  820         INC  DE
64920  830         DJNZ LOOP3
64922  840         POP  DE
64923  850         LD   BC,256
64926  860         EX   DE,HL
64927  870         ADD  HL,BC
64928  880         EX   DE,HL
64929  890         POP  BC
64930  900         DJNZ LOOP2
64932  910         RET
       920 ;Routine to set attributes in the window
       930 ATTR
64933  940         CALL INIT
64936  950         LD   A,(PAPER)
64939  960         RLCA
64940  970         RLCA
64941  980         RLCA
64942  990         LD   B,A
64943 1000         LD   A,(INK)
64946 1010         ADD  A,B
64947 1020         LD   (COLOUR),A
64950 1030         LD   HL,22528: Start of attributes area
64953 1040         LD   DE,32
64956 1050         LD   A,(TOP)
64959 1060         LD   B,A
64960 1070 LOOP4   XOR  A
64961 1080         CP   B
64962 1090         JP   Z,CONT
64965 1100         DEC  B
64966 1110         ADD  HL,DE
64967 1120         JP   LOOP4
      1130 ;HL points to attributes of top line of window
64970 1140 CONT    LD   DE,(LEFT)
64974 1150         LD   D,0
64976 1160         ADD  HL,DE
64977 1170         LD   (POINT),HL
      1180 ;HL points to attributes of top left of window
64980 1190         LD   A,(LINES)
64983 1200         LD   B,A
64984 1210 LOOP5   PUSH BC
64985 1220         LD   A,(CHRS)
64988 1230         LD   B,A
64989 1240         LD   A,(COLOUR)
64992 1250 LOOP6   LD   (HL),A: Colour the character square
64993 1260         INC  HL
64994 1270         DJNZ LOOP6: Back to colour the next square
      1280 ;Top line now coloured in
64996 1290         LD   HL,(POINT)
64999 1300         LD   DE,32
65002 1310         ADD  HL,DE
65003 1320         LD   (POINT),HL
65006 1330         POP  BC
65007 1340         DJNZ LOOP5: Back to colour the next line
65009 1350         RET
      1360 ;Routine to clear the window using repeated scrolls
      1370 CLEAR
65010 1380         CALL INIT
65013 1390         LD   A,(LINES)
```

CROLL

a graphic adventure in which the picture forms part of the permanent display — with the text scrolling continuously in a window underneath it. Unfortunately, the ROM can't help you much here.

On the other hand, perhaps you're looking for a different style of presentation altogether? One (rather controversial) game in the role-playing mould is "Out of the Shadows", in which a small map forms a permanent part of the display, with text continually scrolling in a narrow column up the right hand side of the screen. You really can't attempt this sort of thing in BASIC, can you? There's a lot to be said for trying out unusual methods of on-screen presentation, in fact. There's a tendency for some adventure authors to settle on one particular style and then stick with it, but if this is overdone it isn't long before a certain monotony sets in. The player begins to feel that the the adventure is just another "production-line" effort — and you don't have to look far to find examples. The old "Mysterious Adventures" series had a strong tendency in this direction, and some of the more recent releases by US Gold show all the signs of heading the same way. So it isn't only the Quilled and GAC'd games which fall into stereotyping trap.

Of course we all know that mere appearance isn't the be-all-and-end-all of adventure writing, but there's no doubt that it can help to offset that "seen it all before" impression. Wouldn't it

```
65016         LD   B,A
65017 LOOP7   PUSH BC
65018         CALL SCROLL
65021         POP  BC
65022         DJNZ LOOP7
65024         RET
      ;The first 6 of these addresses are poked from BASIC
65025 TOP     DEFB 0: Top character line of window
65026 BOT     DEFB 21: Bottom character line of window
65027 LEFT    DEFB 0: Left hand column of window
65028 RIGHT   DEFB 31: Right hand column of window
65029 INK     DEFB 0: Window INK colour (0-7)
65030 PAPER   DEFB 7: Window PAPER colour (0-7)
65031 POINT   DEFW 0
65033 CHRS    DEFB 0
65034 LINES   DEFB 0
65035 COLOUR  DEFB 0
      ;The table contains the screen addresses for
      ;the top left hand byte of each line of characters
65036 TABLE   DEFW 16384
65038         DEFW 16416
65040         DEFW 16448
65042         DEFW 16480
65044         DEFW 16512
65046         DEFW 16544
65048         DEFW 16576
65050         DEFW 16608
65052         DEFW 18432
65054         DEFW 18464
65056         DEFW 18496
65058         DEFW 18528
65060         DEFW 18560
65062         DEFW 18592
65064         DEFW 18624
65066         DEFW 18656
65068         DEFW 20480
65070         DEFW 20512
65072         DEFW 20544
65074         DEFW 20576
65076         DEFW 20608
65078         DEFW 20640
65080         DEFW 20672
65082         DEFW 20704
      ;Routine to word-wrap z$ within defined window
      WRAP
65084         CALL INIT
      ;Begin by looking for z$ in variables area
65087         LD   HL,(VARS): Start of BASIC variables
65090 NEXT    LD   A,(HL)
65091         CP   90: z$ identified by code 90
65093         JP   Z,FOUND: Eureka!!
65096         CALL SKIP: Find next BASIC variable (ROM)
65099         EX   DE,HL
65100         JP   NEXT: No luck yet. Try next variable
      ;z$ has now been located in the variables area
65103 FOUND   INC  HL
65104         LD   A,(HL)
65105         LD   (LEN),A: Store LEN z$ (up to 255 only)
65108         INC  HL
65109         INC  HL
65110         LD   (START),HL: Address of start of text
65113 SCAN    CALL CHECK: How long is the string?
65116         CP   1
65118         JP   Z,PRINT: Print it if short enough to fit
65121         CALL CHOP: z$ too long. Prune it
65124         CALL PRINT: Print the pruned portion
65127         LD   HL,(START)
65130         LD   DE,(STEP): No. of chars just printed
65134         LD   D,0
65136         ADD  HL,DE
65137         LD   (START),HL: Start of remaining text
65140         LD   A,(STEP)
65143         LD   B,A
65144         LD   A,(LEN)
65147         SUB  B
65148         LD   (LEN),A: Length of text remaining
65151 REPEAT  LD   HL,(START)
65154         LD   A,(HL): Check next character
65155         CP   32: Is it a space?
65157         JP   NZ,SCAN: If not, process remaining text
65160         INC  HL
65161         LD   (START),HL
65164         LD   A,(LEN)
65167         DEC  A
65168         LD   (LEN),A: Any leading space is removed
65171         JP   REPEAT: Let's see if there are any more
      ;Check length of remaining text
65174 CHECK   LD   A,(CHRS)
65177         ADD  A,1
65179         LD   B,A
65180         LD   A,(LEN)
65183         CP   B: Does remaining text fit window?
65184         JP   C,SHORT: Yes!
65187         XOR  A: No!
65188         RET  ; Return with zero in A
65189 SHORT   LD   A,(LEN): Remaining text not too long
65192         LD   (STEP),A
65195         LD   A,1
65197         RET  ; Return with 1 in A
      ;Prune the remaining text to fit the window
65198 CHOP    LD   HL,(START): Start of text
65201         LD   A,(CHRS): Width of window
65204         LD   E,A
65205         LD   D,0
65207         ADD  HL,DE
      ;HL points to the character at start of next line
65208 CHLOOP  LD   A,(HL): Pick up the character
65209         CP   32: Is it a space?
65211         JP   NZ,DECR: If it isn't, a word is broken
65214         LD   A,E: If it is, we can print this length
65215         LD   (STEP),A: So store length to be printed
65218         RET
65219 DECR    DEC  DE: Back off by one character
65220         DEC  HL
65221         JP   CHLOOP: and try again
      ;Routine to print a line of text
65224 PRINT   CALL SCROLL: Scroll previous text
65227         LD   A,(MASK_P)
65230         PUSH AF: Save old value of MASK_P
65231         LD   A,63
65233         LD   (MASK_P),A: Efectively PAPER 8: INK 8
65236         LD   A,2
65238         CALL #1601: Use the screen
65241         LD   A,22
65243         RST  16: Effectively "PRINT AT:"
65244         LD   A,(BOT)
65247         RST  16
65248         LD   A,(LEFT)
65251         RST  16: Bottom left corner of window
65252         LD   DE,(START): Address of start of text
65256         LD   BC,(STEP): Length of text
65260         LD   B,0
65262         CALL #203C: Print the text (ROM)
65265         POP  AF
65266         LD   (MASK_P),A: Restore old value of MASK_P
65269         RET
 6584 SKIP    EQU  #19B8: ROM routine - find next variable
23627 VARS    EQU  23627: Stores start of variables
65270 LEN     DEFB 0
65271 STEP    DEFB 0
65272 START   DEFW 0
23694 MASK_P  EQU  23694: System variable for transparency

Pass 2 errors: 00
```

be useful to have an off-the-shelf utility which could cater for all likely styles of presentation? For a long time I've tended to take a short term view, writing printing and scrolling routines specifically for the job currently in hand — consoling myself that "one of these days" I'd get around to writing a suitable "all-purpose" utility that could be used for all conceivable situations. Well, "one of these days" has arrived at last; "Multi-Scroll" is about to be launched upon an astonished world.

What we want, essentially, is to be able to define a "window" of any size, anywhere on the screen, so that text can be printed to the bottom of the window and scrolled continuously upwards until it disappears at the top of the window. If we could define INK and PAPER colours for the window, independently of the screen colours elsewhere on the screen, then that would be useful, and it might also be handy to have a simple method for clearing the window completely. Finally, we want to be able to do the actual text printing from BASIC without worrying about whether the individual text lines fit the window, and so we need a suitable word-wrapping routine as well.

Available Services

All these facilitiesare available with Multi-Scroll, and at your service. **Listing 1** is the assembly program — and as always, I strongly recommend that you use this in conjunction with an assembler — but **Listing 2** provides a decimal dump of the machine code (for those who've not yet seen the light). As you'll see, the code is stored from address 64800 onwards, and is 474 bytes long. You'll need a BASIC loader program if you're using the decimal dump, but if you rifle through some back issues of ZXC you'll find something suitable that you can modify. (Don't forget to CLEAR 64799 before you start.)

I've annotated the assembler listing in plenty of detail, so I'll confine my remarks here to a general outline of Multi-Scroll, and how to use it. The machine code consists of essentially four separate routines, each of which can be used from within a BASIC program as required:

1) SCROLL (USR 64825). This scrolls the chosen window upwards by one line. There are several ways of tackling this, and the method I've adopted here uses a table of screen line addresses to minimise the complexities brought about by the Spectrum's eccentric display file. (You can find a more detailed discussion of this approach in Pete Cooke's article in the March issue.)

Machine code data dump.

Listing 2

64800	58	1	254	71	58	2
64805	2	254	144	60	50	10
64810	10	254	58	3	254	71
64815	71	58	4	254	144	60
64820	60	50	9	254	201	205
64825	205	32	253	33	12	254
64830	254	237	75	1	254	6
64835	6	0	9	9	34	7
64840	7	254	42	7	254	94
64845	94	35	86	229	42	3
64850	3	254	38	0	25	235
64855	235	225	58	10	254	61
64860	61	254	0	202	141	253
64865	253	50	10	254	35	34
64870	34	7	254	78	35	70
64875	70	42	3	254	38	0
64880	0	9	6	8	197	237
64885	237	75	9	254	6	0
64890	0	229	213	237	176	225
64895	225	209	1	0	1	9
64900	9	235	9	193	16	234
64905	234	195	74	253	6	8
64910	8	197	213	58	9	254
64915	254	71	175	18	19	16
64920	16	252	209	1	0	1
64925	1	235	9	235	193	16
64930	16	235	201	205	32	253
64935	253	58	6	254	7	7
64940	7	7	71	58	5	254
64945	254	128	50	11	254	33
64950	33	0	88	17	32	0
64955	0	58	1	254	71	175
64960	175	184	202	202	253	5
64965	5	25	195	192	253	237
64970	237	91	3	254	22	0
64975	0	25	34	7	254	58
64980	58	10	254	71	197	58
64985	58	9	254	71	58	11
64990	11	254	119	35	16	252
64995	252	42	7	254	17	32
65000	32	0	25	34	7	254
65005	254	193	16	231	201	205
65010	205	32	253	58	10	254
65015	254	71	197	205	57	253
65020	253	193	16	249	201	0
65025	0	21	0	31	0	7
65030	7	0	0	0	0	0
65035	0	0	64	32	64	64
65040	64	64	96	64	128	64
65045	64	160	64	192	64	224
65050	224	64	0	72	32	72
65055	72	64	72	96	72	128
65060	128	72	160	72	192	72
65065	72	224	72	0	80	32
65070	32	80	64	80	96	80
65075	80	128	80	160	80	192
65080	192	80	224	80	205	32
65085	32	253	42	75	92	126
65090	126	254	90	202	79	254
65095	254	205	184	25	235	195
65100	195	66	254	35	126	50
65105	50	246	254	35	35	34
65110	34	248	254	205	150	254
65115	254	254	1	202	200	254
65120	254	205	174	254	205	200
65125	200	254	42	248	254	237
65130	237	91	247	254	22	0
65135	0	.25	34	248	254	58
65140	58	247	254	71	58	246
65145	246	254	144	50	246	254
65150	254	42	248	254	126	254
65155	254	32	194	89	254	35
65160	35	34	248	254	58	246
65165	246	254	61	50	246	254
65170	254	195	127	254	58	9
65175	9	254	198	1	71	58
65180	58	246	254	184	218	165
65185	165	254	175	201	58	246
65190	246	254	50	247	254	62
65195	62	1	201	42	248	254
65200	254	58	9	254	95	22
65205	22	0	25	126	254	32
65210	32	194	195	254	123	50
65215	50	247	254	201	27	43
65220	43	195	184	254	205	57
65225	57	253	58	142	92	245
65230	245	62	63	50	1,42	92
65235	92	62	2	205	1	22
65240	22	62	22	215	58	2
65245	2	254	215	58	3	254
65250	254	215	237	91	248	254
65255	254	237	75	247	254	6
65260	6	0	205	60	32	241
65265	241	50	142	92	201	0
65270	0	0	0	0	0	0

CHECKSUM 67338

Demo program.

Listing 3

```
  1 REM ***DEMONSTRATION PROGRAM***
 10 CLEAR 64799: LOAD *"m";1;"scrollCODE "CODE 64800
 20 BORDER 0: PAPER 0: INK 7: CLS
 29 REM ***Assign addresses to suitably named variables***
 30 LET top=65025: LET bot=65026: LET left=65027
 40 LET right=65028: LET ink=65029: LET paper=65030
 50 LET scr=64825: LET attr=64933: LET clr=65010: LET prt=65084
 59 REM ***First window***
 60 GO SUB 500
 69 REM ***First message***
 70 LET z$="Here is a wide window at the bottom of the screen.
Press a key to scroll it up one line.": RANDOMIZE USR prt
 80 GO SUB 9000: RANDOMIZE USR scr: REM *(Single scroll)*
 89 REM ***Second message***
 90 GO SUB 9000: LET z$="Next we'll create some more text windo
ws. First, we'll have one on the right.": RANDOMIZE USR prt
 99 REM ***Second window***
100 GO SUB 9000: GO SUB 600
109 REM ***Third message***
110 LET z$="Here we have a tall, narrow window. You could use t
his to scroll text up the side of the screen with some sort of g
raphics on the left. This is the kind of display used by games s
uch as ""Out of the Shadows"".": RANDOMIZE USR prt
120 GO SUB 9000
129 REM ***Third window***
130 GO SUB 700
139 REM ***Final message***
140 LET z$="Finally we'll clear each of the windows one by one.
Ready?": RANDOMIZE USR prt
150 GO SUB 9000
159 REM ***Clear first window***
160 GO SUB 500: RANDOMIZE USR clr: GO SUB 9000
169 REM ***Clear second window***
170 GO SUB 600: RANDOMIZE USR clr: GO SUB 9000
179 REM ***Clear third window***
180 GO SUB 700: RANDOMIZE USR clr: GO SUB 9000
200 GO TO 60
499 REM ***Long window at screen bottom***
500 POKE top,18: POKE bot,21: POKE left,0: POKE right,31: POKE
ink,1: POKE paper,6: RANDOMIZE USR attr
510 RETURN
599 REM ***Tall window at right hand side***
600 POKE top,1: POKE bot,15: POKE left,18: POKE right,31: POKE
ink,6: POKE paper,2: RANDOMIZE USR attr
610 RETURN
699 REM ***Little window on left hand side***
700 POKE top,5: POKE bot,13: POKE left,0: POKE right,6: POKE in
k,2: POKE paper,7: RANDOMIZE USR attr
710 RETURN
8999 REM ***Wait for keypress***
9000 PRINT #1; INVERSE 1;AT 1,5;"PRESS A KEY TO CONTINUE": PAUSE
 0: PRINT #1;AT 1,0,,: RETURN
```

2) ATTR (USR 64933). This washes the window with the chosen INK and PAPER colours. Note that it doesn't *clear* the window — it only changes the colours for whatever is already there, and for any subsequent printing.
3) CLEAR (USR 65010). This performs as many scrolls as there are lines of characters in the window; in other words, everything previously printed in the window will be scrolled upwards and off the top, leaving the window clear.
4) WRAP (USR 65084). Regular readers will have encountered similar routines to this in one or two of my previous articles but we require greater flexibility here, and this one represents (I hope!) my last word on the topic. Its purpose is to print text held in the BASIC variable z$ in the window, scrolling the lines and word-wrapping as necessary — so you don't need to worry about padding your text with spaces to make it fit the window; this routine will sort all that out for you. The only limitation is that LEN z$ should not exceed 255 (which means that for really long chunks of text, you'll need to break it into separate sections — and you'd probably do this anyway to avoid textual claustrophobia on-screen.)

This, then, is what Multi-Scroll does — but how do we control it from BASIC? Very simple. There are six addresses whose contents determine the behaviour of the routines, and these should be POKED with suitable values before the routines are used. Four control the size and position of the window, and two control the colours as follows:

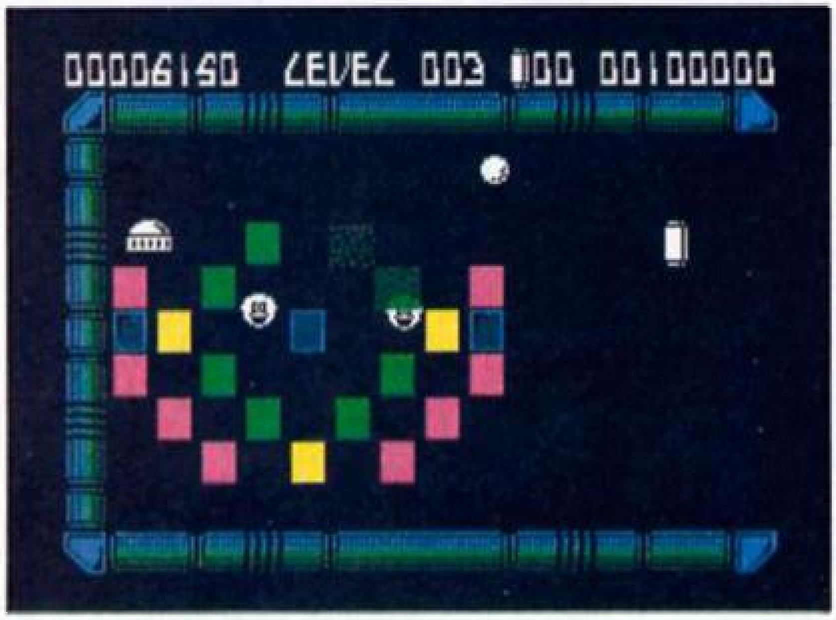
1) The limits of the window are defined using the four addresses from 65025 to 65028 inclusive (labels TOP, BOT, LEFT, and RIGHT respectively — see Listing 1). Think in terms of the Spectrum's normal system of identifying character squares, and set the limits accordingly. For example, if you POKE 65025,19: POKE 65026,21 then your window will be three lines high (with line 21 the bottom line to which printing is done). POKE 65027,0: POKE 65028,31 will set the left and right limits so that your window will occupy the whole width of the screen (from character square 0 to character square 31). Incidentally, no check is made for invalid screen positions within the routines, so don't go poking any old numbers in here!

2) The colours (from 0 to 7 inclusive, as usual) are defined using addresses 65029 and 65030 (INK and PAPER respectively). Again, no check is made for invalid colours.

Other uses

Although the most obvious use for Multi-Scroll is in a graphic adventure game, with text scrolling continuously below an illustration of some kind, there's nothing to stop you from using it in other contexts, perhaps with several windows on screen at once, by setting up a small number of BASIC subroutines to define them. To show how this can be done, and to illustrate all the things we've been talking about, I've given in Listing 3 a short demonstration program which will put Multi-Scroll through its paces for you. Notice that the various USR call and POKE addresses are allocated to suitably named variables in lines 30 to 50; this makes for improved readability, and saves you looking up the numbers each time. Save the demonstration program to autorun from line 1 — if you're working from tape rather than microdrive, you'll need to change the LOAD instruction in line 10, and save the machine code block immediately after the BASIC. If you follow what's happening on screen, comparing it carefully with the listing, you should rapidly get the hang of things. Have fun ...

GAMES



KRAKOUT
Gremlin
£7.95

This is another Breakout variant following soon after Ocean's Arkanoid tried to breathe new life into the old formula. Arkanoid succeeded but unfortunately the same cannot be said of Krakout whose new additions have left it devoid of the appeal of the original game.

Like Arkanoid there's a poor attempt at a plot. In this case you are trying to "out manouevre the awesome ogre". Of course to accomplish this you must blast your way through wall upon wall of bricks.

The additions to the game include expanded bat size, glue bat (where the ball sticks to your bat) missiles, bombs, extra lives etc. These are actifvated if you succeed in hitting the right block but as you have little control over which block you hit once the ball is on its way, it's all fairly arbitrary. In Arkanoid the activating method was catching falling capsules which added to the complexity of the game.

Another change is that Krakout is played horizontally across the screen rather than the familiar vertical play of Breakout. I can't really see any advantage in this change, in fact it seems to detract from the gameplay.

On the plus side the graphics are very bright, you can vary the speed of the game and there are 100 screens worth of assorted bricks to batter, but in the final analysis it lacks an enduring addictive quality.



SIGMA 7
Durell
£7.95

Described as an "all action arcade game featuring imaginative state-of-the-art 3D graphics and exciting synthesised sound", this is actually three reworked examples of some old games.

The first phase of Sigma 7 is to fly through space to the factory while fighting off the space mines that swarm around you in a series of attack waves. These mines look remarkably similar to alien spacecraft and even fire missiles at you. After surviving a below standard shoot em up you arrive at a factory that introduces the latest version of Pacman!

In the factory you must clear the pathways by collecting dots while being chased by robomines that home in on you from all parts of the scrolling maze. These can be shot with your laser but your main objective is to find the pattern of dots that can't be moved as this forms a combination for stage three.

The third and final stage is a futuristic puzzle in which you must enter the pattern onto a control box while avoiding a killer sphere that tries to touch your cursor and kill one of your lives.

You begin the game with the standard three lives but gain another three lives when you complete a stage. You could have nine lives by the time you've finished the first level and you'll need them to survive later levels when you have to face more space mines, clear a harder and bigger factory maze and enter a more complicated code.

The graphics are good and do add a new lease of life to these tried and tested game formats but for £7.95 you expect something different.





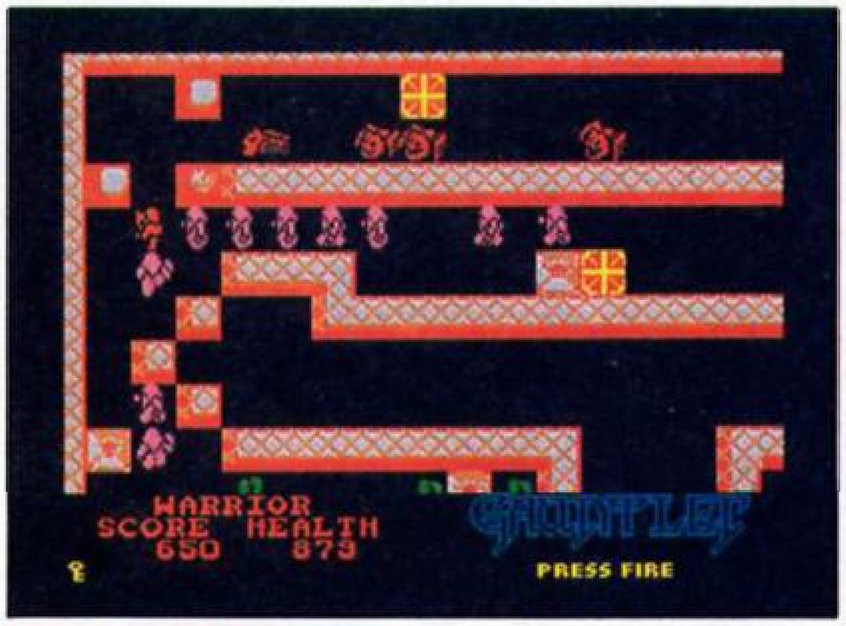
BRAINACHE
Code Masters
£4.99

Brainache's done it again. He's left all his mining equipment in the Stella mines and now he's got to back, and get it, urged on by the rest of the crew's laughter. This time he's determined to show them all how clever he really is and decides to delve deeper into the mines and search for the fabled Anastasie diamond.

Although the game features the most stunning backgrounds you are likely to see in a budget game, the game itself is awful. Our hero is little more than a stick man that smudges badly into the background as do all the flying, crawling and plantlike nasties that are out to get you. Hidden in the ground are the platforms and slopes that you can walk on and plummet off as the screen jerks and shakes in its attempt to scroll. Forget Brainache, this one's just a pain.







GAUNTLET — THE DEEPER DUNGEONS

US Gold
£4.95

So you rushed out and bought Gauntlet didn't you and spent a month of sleepless nights raiding its 512 levels of food, potions, keys and treasure but now all the monsters are dead! What are you going to do?

You've played some of the clones but they don't quite sooth the itch to wipe out grunts, ghosts, sorcerors and lobbers. Don't worry as US Gold have just released the Deeper Dungeons featuring 512 additional levels or monsters, magic and mayhem.

The dungeons were created by the players of the original Gauntlet who sent in their designs to create this compendium of chaos. Since the levels were created by players they are actually harder and more involved than the original and so you should find it a whole new challenge.

Ghosts and grunts are soon joined by the more deadly sorcerors, lobbers and deaths as the game starts at top speed. In fact you have to blast your way through a section of wall in the second level, most potions are guarded by deaths (only destroyed by potions!) and level 4 begins with a shower from rock throwing lobbers!

The Deeper Dungeons are loaded into the original game (which you obviously must have!) when it tells you to load in side 2.

The graphics and game-play are just as quick and impressive as the original which won't disappoint Gauntlet addicts but for a sequel I would have liked the inclusion of some new monsters and magic artefacts to collect. Who knows, perhaps there's an "Even Deeper Dungeons" waiting in the wings.



TRAP

Alligata
£8.95

I got off to a good start with Trap. I loaded it up, left a bit in the dark by the uninformative cassette inlay, and found myself unable to get the game started after selecting joystick control. The instructions weren't any help, so I ended up thumping away at the keyboard trying to find the right key. Once I got going I found that the game doesn't seem to respond to the 128/+2's own Sinclair interface so I had to unplug and start loading again, this time with a Kempston interface (again, there's no hint of this in the instructions).

I eventually got the whole thing loaded and working, and came face to face with a fairly passable shoot 'em up. Trap is very much in the mould of most space zap games since the release of Lightforce. You start off with a short space flight in which you have to dodge or destroy waves of space mines. Then, as you reach your destination you fly over a valley in which you have to destroy lasers on the ground and boats in the river below, as well as combating alien forces in the air. The third part of the game has you on foot, attempting to collect orbs (which represent currency/points of some sort).

Once you have completed these tasks you go through them again, though this time at a higher difficulty level and with the option of selecting larger, more powerful ships of your own depending on your score in earlier games.

It's all quite professionally done, but a bit lacking in originality. *I know that you're* meant to judge each game on its own merits, but like so many recent shoot 'em ups, Trap reminds me of Lightforce but without the same level of high-speed addictiveness.





TRANSMUTER

Code Masters
£1.99

As the Sun grew old and turned into a red giant the ancient Earth culture retreated into the core of the planet leaving a series of formidable defences in their wake. Aeons later the Overlord of Tenz (that's you!) returns to Earth in a one off total aggression ship to destroy the old defences and clear the way for safe space travel.

Your ship is not only equipped with a standard photon cannon but also a nucleonic Transmuter. This definitely non-standard weapon sucks in debris from around the ship and converts this into pre-programmed forms. So as the game proceeds and you blast away at the Earth defences your ship gets stronger!

The game is set in the caverns that lead to the Earth's core and are packed with vector graphic nasties such as limpet guns that blast you, tops that float just out of the path of your cannon bolts and generators and pour nasties into your path.

Later on you'll have to face combinations of limpet guns, trundling tanks and generators as the *cavern walls narrow*. As you blast these these you'll be offered various additions as the Transmuter recycles the debris to produce extra speed, double shot (you fire six bolts instead of three at a time), a laser (instantaneous bolt), shield (absorbs one collision) and a second cannon that appears below the ship but makes it easier to hit. Unfortunately you can only have one of these at once but since the menu cycles through them as you take out nasties you should be able to get what you want.

Complete the first cavern and you'll get a chance at a bonus screen of blasting ships in space (dead easy) and then on to the next cavern that's totally nasty free! The only trouble is that you have to steer your ship through a maze of scrolling, winding caverns where the slightest mistake will cost you a life.

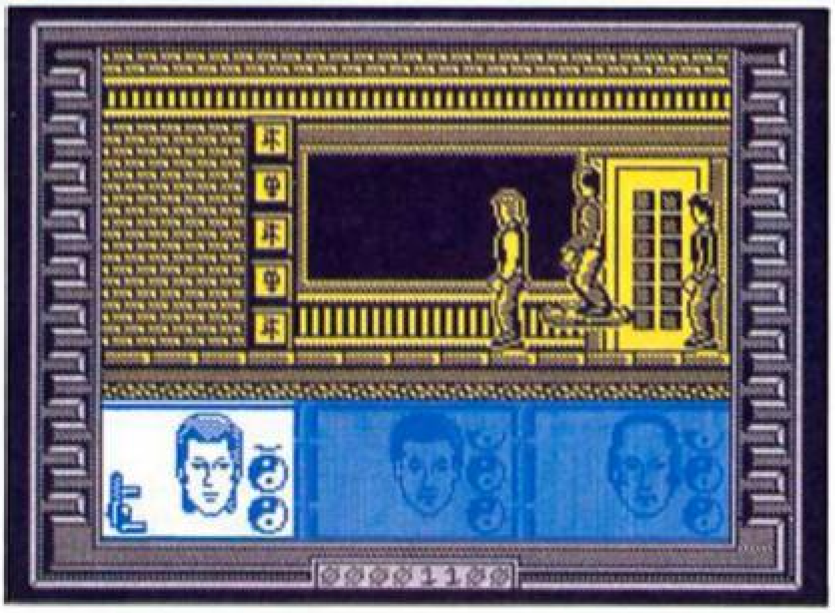
An excellent game that shows that the old Scramble format is still alive and kicking. For only £1.99, what more could you *want. Sheer budget brilliance!*





BIG TROUBLE IN LITTLE CHINA

Electric Dreams
£9.95

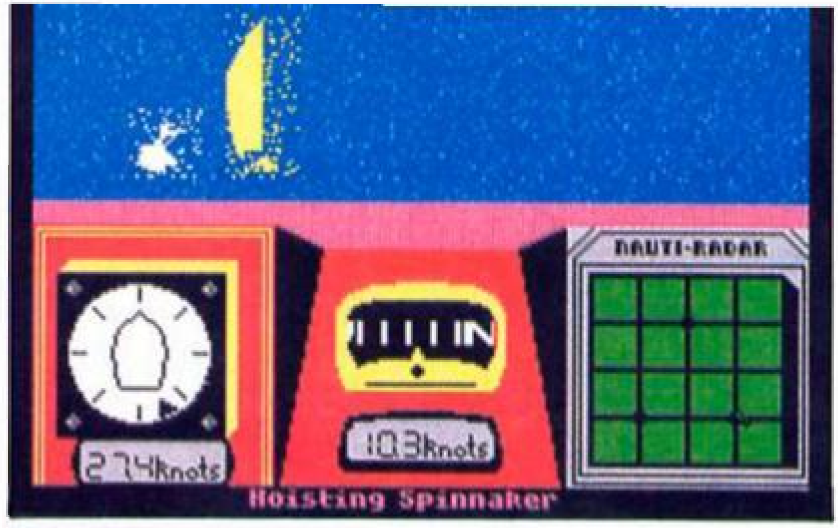


The problem with licensing deals is that they are worked out so far in advance that nobody knows whether the original film is going to be a success. Electric Dreams seems to be buying film licences in job lots and Aliens was certainly a winner. Big Trouble in Little China, however, was released and disappeared without trace. Given that there aren't too many people who will be reliving the film through the game, Big Trouble in Little China has to stand entirely on its own merits. Unfortunately the game is little more than a run of the mill kung fu game with three characters with various combat skills making their way through four levels of action and beneath the streets of Chinatown, to rescue their girlfriends from the clutches of Mandarin Lo Pan.

Each of the three characters are specialists. Jack Burton is a marksman and there is a gun to be found once you've penetrated into the game. Wang Chi is a karate expert and skilled at sword play. Egg Shen is the most bizarre character as he flies around on a cloud hurling thunderbolts. As you progress through the levels you are confronted with both humans and monsters to defeat. You can switch control between characters very easily but as they are not independent characters and do nothing while they are uncontrolled, it seems an option whose potential is wasted.

The action simply scrolls from right to left and the same scenery tends to crop up regularly giving the impression you are just going round in circles. The game action is not particularly compelling and the fight sequences are no more than average. Although there is nothing glaringly bad about Big Trouble there is nothing new or exciting either to set it apart from the general run of releases.

GOOD



SAILING

Activision
£9.95

If you thought that the America's Cup would be won and lost and then racing would return to obscurity then you'd be wrong, as the software houses have always been keen to climb aboard a bandwagon. Unfortunately, this floating bandwagon isn't likely to stay afloat for long because yacht racing isn't the most exciting of spectator sports. Even during the hype of the America's Cup, live TV coverage failed to catch on as the races took over half an hour, even though they were often decided by the first marker.

Activision has attempted to liven up the sport by shortening the races (around 3 instead of 8 markers), allowed you to design and redesign your boat as well as simplifying the actual sailing. The result is a highly playable game in which you compete against an international field of 17 other yachts in a ladder league.

After an all-important weather forecast the week's racing begins, in which you can race any yacht just above you in your custom designed craft.

During the races the screen display shows an impressive 3D view of the course and the other yacht if he's in the lead, and a map of the course, speedometer and wind direction indicator.

The secret of how to get around the course lies in when you hoist and stow your giant spinnaker sail. This can double your speed with the wind behind it, but stop if you get it wrong.

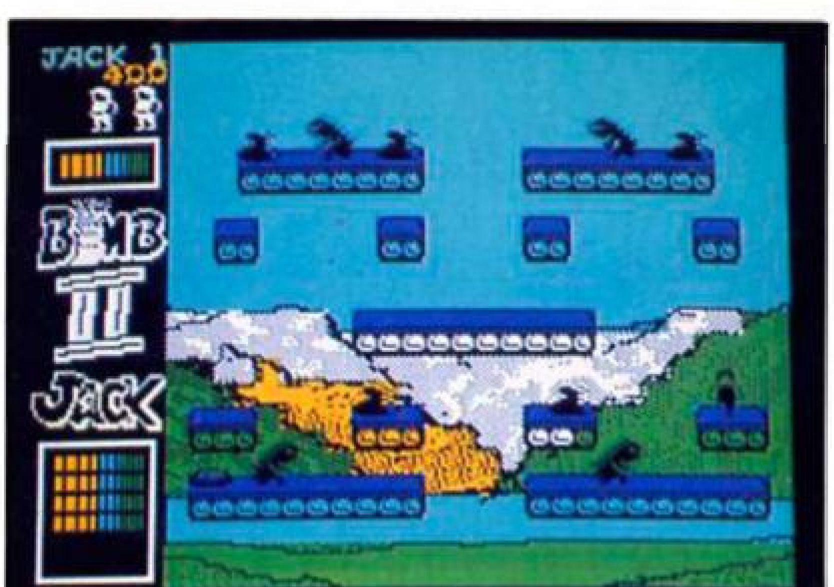
Since the course is triangular this is a problem you'll have to solve to climb the ladder to the championship.

The actual racing is only half the contest as you also have to design your boat and decide such things as the boat length, waterline length and keel design. The result is a game that's a lot more exciting than the sport it simulates but unlikely to win over new fans.

GOOD

BOMB JACK II

Elite
£7.95



Well that's it, the end of an era. Bomb Jack II arrived in the office and once I'd clubbed all the competition senseless I won the honour of loading it up and getting to review the sequel to Bomb Jack, one of the most addictive games in the history of the entire universe. But to be honest, it was a bit of a disappointment.

The basic format of the game is more or less the same as in the original arcade game — you guide Bomb Jack around dozens of screens full of platforms and bombs, in an attempt to defuse them all — but somehow the 'feel' of the game isn't the same. The main difference between the two games is that in BJ II, you now have a 'stab' button that allows you to fight the reptiles on the earlier screens. Somehow though, I don't think that this really adds to the game since it actually removes some of the element of high speed panic that was so much a part of the first game. This does help if you're trying to rack up a high score but isn't as much fun as all that leaping around and trying to defuse the bombs in the correct order at the same time as avoiding the monsters that are patrolling the platforms.

I would have expected the graphics to be tidied up a little too. The original Bomb Jack suffered from some quite bad masking problems when sprites overlapped. These have been improved, but the graphics now look a little bit cramped and the figure of Jack looks a bit like a small man in a flat cap rather than the dynamic caped sprite that we're all use to.

To be fair, BJ II isn't that bad a game but I simply didn't find it as compellingly addictive as its predecessor. On the other hand, the original Bomb Jack is included on the other side of the tape so if you haven't already got it, then it's probably worthwhile buying Bomb Jack II and getting both games for the price of one.

GREAT

COMPETITION

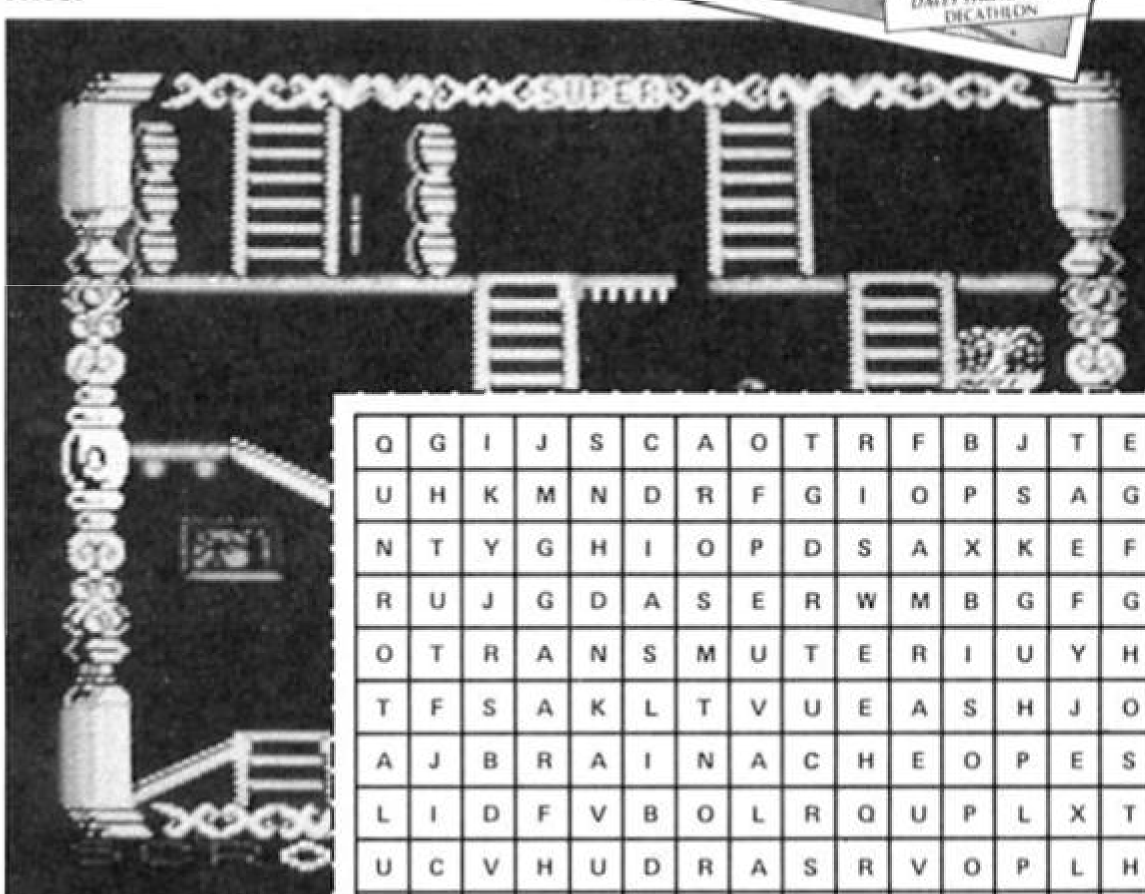


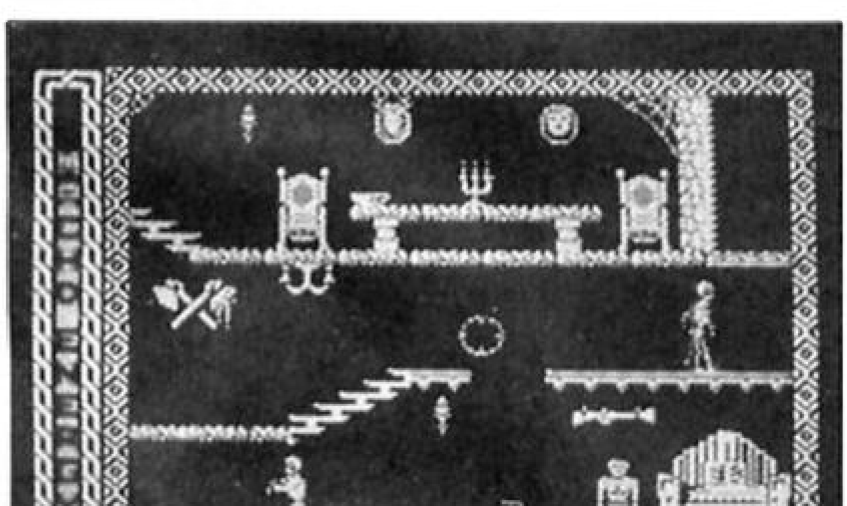
After ploughing our way through all the recent budget software releases we thought it might be a good idea to round up some of the best of them and offer them as prizes in a competition. Imagine our surprise when it turned out that most of the titles we thought of were by Codemasters, so here we are with a special Codemasters competition offering 12 sets of their four latest games, Star Runner, Super Robin Hood, Ghost Hunters and Transmuter.

Win a set of Codemasters' budget hits.

How to Enter

All you have to do to enter the competition is to find the names of those games and two other recent Codebusters' releases in the word square below. Then just fill out the details of your name and address and send it in. All entries must arrive by first post on the 10th July 1987. The competition is open to all readers of ZX Computing except employess of Argus Specialist Publication, Codemasters Software Co and Chase Webb. The editor's decision is final and no correspondence can be entered into.





Q	G	I	J	S	C	A	O	T	R	F	B	J	T	E
U	H	K	M	N	D	R	F	G	I	O	P	S	A	G
N	T	Y	G	H	I	O	P	D	S	A	X	K	E	F
R	U	J	G	D	A	S	E	R	W	M	B	G	F	G
O	T	R	A	N	S	M	U	T	E	R	I	U	Y	H
T	F	S	A	K	L	T	V	U	E	A	S	H	J	O
A	J	B	R	A	I	N	A	C	H	E	O	P	E	S
L	I	D	F	V	B	O	L	R	Q	U	P	L	X	T
U	C	V	H	U	D	R	A	S	R	V	O	P	L	H
M	I	P	Y	T	F	C	E	S	D	U	B	A	W	U
I	H	G	F	S	W	R	B	N	I	O	N	E	R	N
S	Z	X	U	E	D	G	H	O	S	W	E	N	O	T
X	B	R	A	I	N	T	R	A	B	O	X	Y	E	E
M	O	R	T	D	I	P	T	L	E	V	G	U	S	R
B	D	O	O	H	N	I	B	O	R	R	E	P	U	S

Name. . . . .

Address. . . . .

. . . .

. . . .

Complete this coupon and send it to:- Codemasters Competition, ZX Computing Monthly, 1 Golden Square, London W1R 3AB. Closing date is 10th July 1987.

Return of the Artist

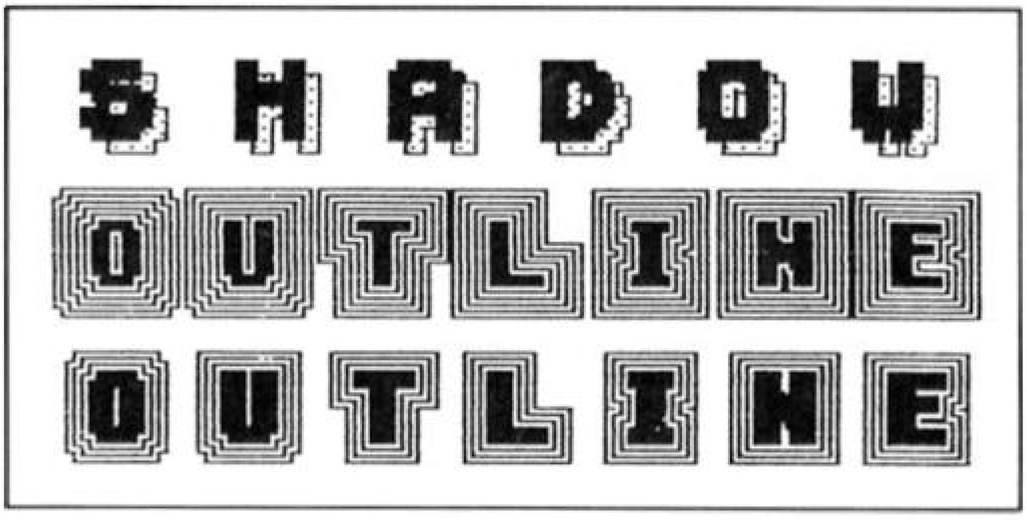
Carol Brooksbank reviews the new 128 version of Artist II, and the debugged 48K.

Good news from Softek! You may remember that when I first reviewed THE ARTIST II in February, I commented on the bugs in the program and said that Softechnics really should take another look at it. Well, I am pleased to report that they have done just that. The 48K version has been re-issued without the bugs, and is now the really splendid program it always promised to be.

It has all the screen drawing facilities that you would expect — elastic lines, filled or unfilled shapes, alternative typefaces, freehand drawing, keyboard, joystick or mouse control, enlarged areas for detailed work, font and sprite designers, numerous and re-definable fill patterns. But the three features which lift this program head and shoulders above the opposition are the window option, cut and paste, and the Pagemaker.

The window option allows a rectangular window, of any size from one character square to the full screen, to be defined. The graphics within it can then be manipulated — turned, inverted, enlarged or reduced, copied to another part of the screen, mirrored, outlined or thickened. The window can be scrolled, and attributes reset. But the really exciting feature is the insert facility. A second screen can be loaded into memory, then scrolled around in the window area, until the required section is within the window. The graphics from the second screen are then copied into the current artwork. The window option only allows areas corresponding to the character squares to be manipulated, and movement about the screen is in character square jumps. To cut irregular shapes, or work to pixel accuracy, the cut and paste option is used.

Here you can draw a box, circle or irregular shaped line around the exact section you wish to work on, and then move it to the new location, turn, mirror or invert it. It can be ANDed, XORed or ORed into the new position. This option has the same insert facility as the window feature, so that pieces can be cut from a second screen and inserted into the current artwork. The scroll option in cut and paste moves in pixel steps, so that the final placement can be absolutely precise. The illustration of typefaces shows some of the effects available by using these options.



Pagemaker

The third unique feature of Artist II is the most exciting of all. This is the Pagemaker, a page layout facility, which enables you to combine text and graphics and print an A4 page. The text, which must be produced using THE WRITER word processor, forms the 'skeleton' of the page, which can hold 96 lines by 64 columns of text, or up to eight full screens of graphics. A full screen occupies 24 lines x 32 columns. When the text is being typed, spaces must be left where the graphics are to be inserted. Any size illustration, from 1 character square to a full screen, can be used so the layout must be carefully planned at the word processing stage. There are five type faces available which are selected by inserting control codes into the text, as are underlining or inverting. The finished text is saved, and then loaded into the Pagemaker.

Pagemaker divides the page into four blocks, each capable of holding two full screens side by side. You can scroll across the block to read the full width of the page. A graphics screen is loaded into memory, and a window defined on the block where the illustration is to be placed. The graphics are then scrolled to bring the correct portion into the window. When all the illustrations in the topblock are in place, it can be saved and/or printed, and the procedure repeated for each of the other four blocks. The resulting page is very attractive, (see illustration).

If you wanted to print multiple copies of a many-paged document, this could all be a bit tedious, especially if you save to tape, because four separate blocks must be loaded and printed one after the other

Some of The Artist II's typeface effects.



to produce each page. However, if you are using microdrive or disc, a FOR-NEXT loop could easily be written to print several copies of a page, loading each block and printing it automatically.

The illustration shows the other application of the Pagemaker. If you want to make a large illustration, covering several screens, it is virtually impossible to print several ordinary screen dumps on the same page and line them up with the pixel accuracy necessary to make a single picture. Several dumps pasted together had been my solution until now, but this program makes it possible to put up to eight screens together accurately and produce one printout.

The handbook says that the printout will fill the page, and any margins must be left within the text area. However, with my combination of Kempston E interface/Epson RX80 printer, this is not so. There is just the right margin on the paper using the full 64 columns, but the problem is that it is all on the right hand side of the page. I discovered that this can be overcome if you go into BASIC, poke 64005-64010 (inclusive) with 0, and enter.

LPRINT CHR$ 27; "1"; CHR$ 6;

as a direct command. (Make sure this is the Pagemaker you are working in, not the main program, before you do this!) The illustration shows the layout this gives you.

If you are looking for a 48K graphics package, then my advice would be to look no further than this one. The new 128K version is exactly the same as the 48K, except for the Pagemaker, which is even more powerful.

128

In the 48K package, Pagemaker is a separate program. In the 128K version it becomes an option of the main program, accessed from one of the pull-down menus. You no longer work on the page in four separate blocks. The whole page is available and the screen is scrolled around it to view various sections. The text is prepared in the same way as for the 48K version, but the graphics screen holding the illustration is loaded into the main program. A whole screen can be inserted straight into the page, but where smaller portions are to be used, a screenful of the Pagemaker page is 'grabbed' into the main program, and the window and cut and paste options used to insert the illustrations. The portion is then replaced into the Pagemaker.

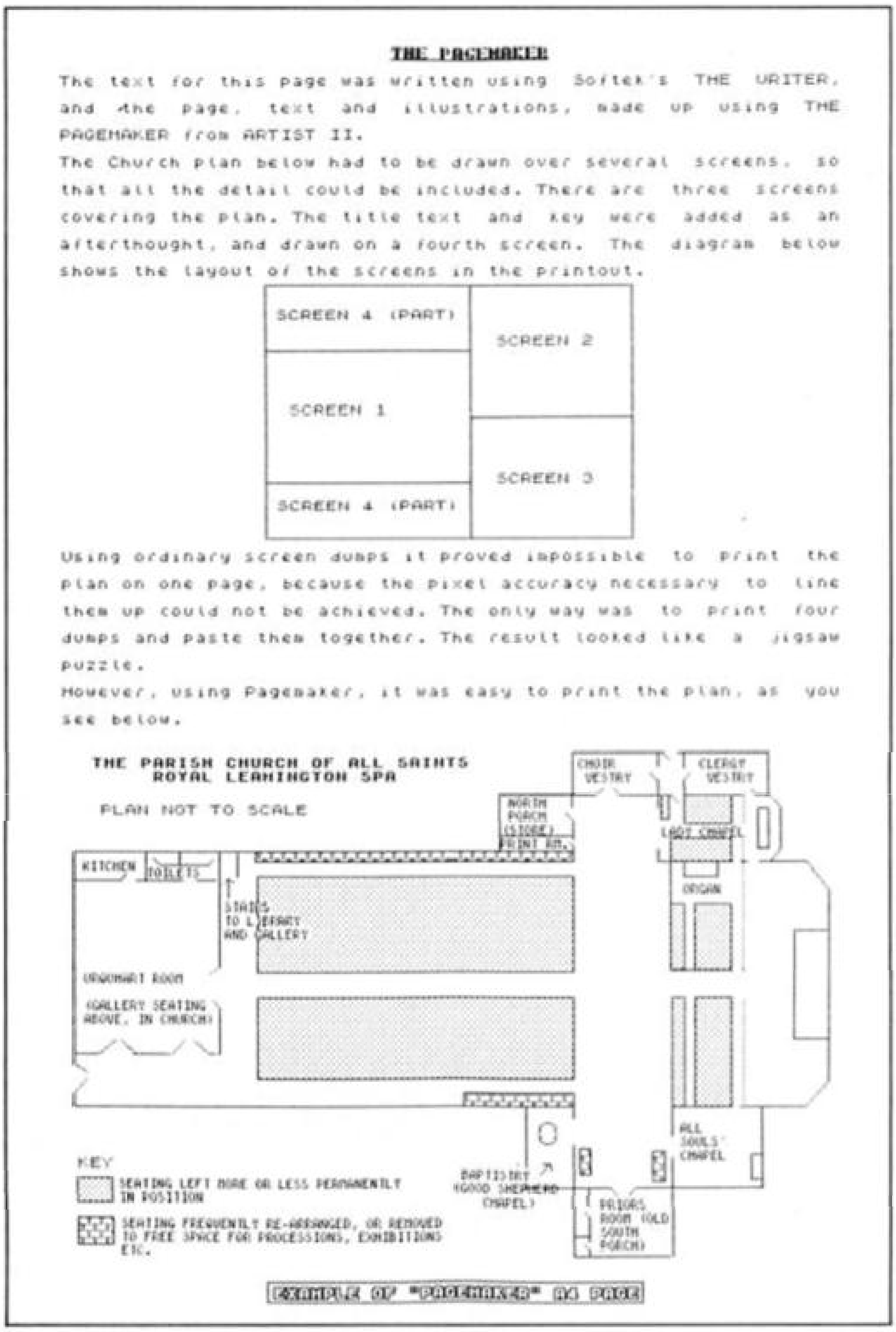
In the 48K Pagemaker you can only bring prepared text and illustrations together, but the 128K version allows you to modify both text and illustrations, perhaps enlarging header text to make a 'headline', modifying an illustration or even drawing a new illustration direct into the page. There is a useful preview option, which gives you an overview of the page layout. It is too small to read the text or reproduce the illustrations properly, but the blocks of text and illustration shapes are shown. You can see whether, for instance, you have centred an illustration on the page, which can be quite difficult to judge when scrolling around the full size page.

The whole page is saved and printed automatically in the 128K version, which makes it a much more viable proposition for 'desk top publishing', though it would still need a patient soul to contemplate producing 100 copies of a 20 page document by this method. But for a leaflet, or a short report, and a limited number of copies, it is a great step forward for the Spectrum.

If, like me, you use a Kempston E interface you cannot, of course, print from 128K mode. Fortunately, the full page is saved in the four blocks which are compatible with the 48K version, so I find it worthwhile, since I do a fair bit of this sort of work, to own both versions. I prepare the page in 128K, taking advantage of the extra facilities, save it, and then print from 48K.

Even if you do not wish to produce a printed page, the ability to hold eight full screens in memory at once, and scroll around them freely, will be very useful to anyone preparing screens for a games program.

The final verdict has to be that this is the most advanced graphics package available for the Spectrum. Its compatibility with THE WRITER, with the promise of THE FILER to be coming shortly, must surely mean that this will be the top Spectrum business suite.



THE PAGEMAKER

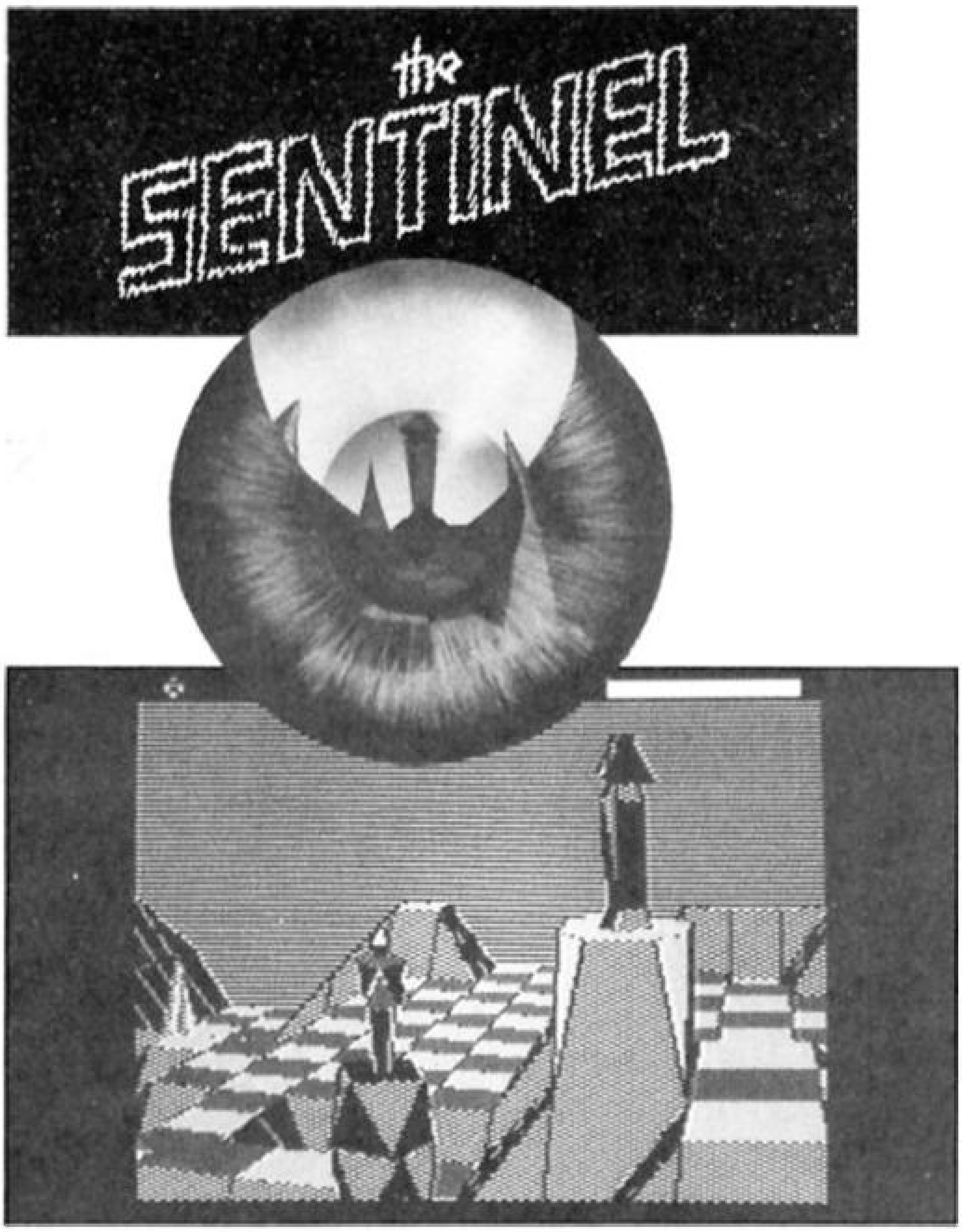
The text for this page was written using Softek's THE WRITER, and the page, text and illustrations, made up using THE PAGEMAKER from ARTIST II.

The Church plan below had to be drawn over several screens, so that all the detail could be included. There are three screens covering the plan. The title text and key were added as an afterthought, and drawn on a fourth screen. The diagram below shows the layout of the screens in the printout.

Using ordinary screen dumps it proved impossible to print the plan on one page, because the pixel accuracy necessary to line them up could not be achieved. The only way was to print four dumps and paste them together. The result looked like a jigsaw puzzle.

However, using Pagemaker, it was easy to print the plan, as you see below.

EXAMPLE OF "PAGEMAKER" A4 PAGE



The Sentinel
Firebird
£9.95

An intriguing and original release from Firebird

In a land far away where pure energy is the only source of power the Sentinel rules supreme. Now you aim to challenge this master in a battle that will rage over 10,000 landscapes that are amazingly crammed into 48K!

Each landscape is a world of its own built out of plateaus, peaks and pits. On top of the highest peak stands the Sentinel. You begin at the bottom of the deepest pit.

The sentinel from it's lofty throne slowly rotates and scans the landscape beneath it for anything with a surplus of energy, such as you. The Sentinel absorbs the energy of living things and recycles it into the lowest life form which is a curious conical tree that it scatters around the barren landscape. You can absorb these trees to collect the energy to defeat the Sentinel.

You are in fact a robot that can't move but can duplicate robot shells from energy and transfer to them. You can then reabsorb the old robot shell thereby reclaiming the energy. Using this method you can move to any plateau you can see below you, and can rise up levels by building boulders on top of plateaus then the robot shell on top of that. You can stack boulders to any height (although you can only see the top of one boulder) and build upto the highest level. However, a boulder costs two units of energy and a robot three which tends to limit your excursions. If you think you're in desparate trouble you can hyper space (costs three units) but you'll end up at a equal height or even lower plateau.

If the Sentinel traps you in its deadly gaze it will gradually absorb your stored energy until you either transfer to another robot, hyperspace or die. If it spots you but can't see the base of the plateau you're standing on it will try to flush you out by converting the nearest tree into a meanie. These snake like creatures also rotate and if it sees you before you can absorb it or get away then it'll force you to hyperspace. This will cost you three units and will probably land you in the gaze of the waiting Sentinel.

By landscape thirteen things are getting a little trickier as the Sentinel has now recruited the help of a Sentry. This is a mini Sentinel that also stands on a tall peak and absorbs anything caught in it's gaze, creates Meanies to flush you out and of course guards the Sentinel. In later landscapes the Sentinel recruits more and more Sentries which must be destroyed before you can tackle the Sentinel. By now the landscape is also getting flatter with less cover in which you can hide from the energy draining gazes of the Sentinel and his Sentries. In one landscape you begin in the centre of a wide open circle surrounded by nine peaks containing the Sentinel and eight Sentries!

The game is a mixture of tactics, energy collecting and reactions as you escape Meanies and the Sentinel. Your first task is to find the Sentinel (for Sentry) before they find you and gauge the direction and speed that it's rotating. You can then aim to stay on its blind side while you absorb enough trees for your assault. To absorb the Sentinel and complete the landscape you must get above it (by standing on boulders and absorb it by homing in on it's base. One landscape down only 9,999 to go!

You don't actually have to complete every landscape as on completion of one you're given the eight digit code for your next one. The next code you're given depends on the ammount of energy you have left once the Sentinel has been defeated. The more energy you have the more landscapes you skip. Soon you will have pages and pages of eight digit codes corresponding to the games landscapes. This means you can start any game at any landscape you like as long as you have the code. It's best to keep all the codes you find as you may get stuck on a particularly tough level that you could leap frog by redoing a previous level and aim to finish it with more energy.

The Sentinel is an absorbing mix of gameplay demanding strategy and skill to solve it. Original, addictive and massive, what more could you possibly want?

MONSTER HIT

Throne of Fire

Throne of Fire
Melbourne House
£7.95



The latest game from the author of Doomdark's Revenge.

In between his efforts to complete Dark Sceptre, Mike Singleton has managed to find time to design Throne of Fire for Melbourne House. However, the design is all that he's done, for the inlay credits all the programming, graphics and music to Consult Computer Systems. Perhaps this rather piecemeal approach explains why the game doesn't quite seem to know what it wants to be.

The action takes place within the Burning Citadel, home of the King of Carakesh who has just shuffled off this mortal coil, and centres around his three sons. Alorn (well meaning but thick), Cordrin (the good guy), and Karag (boo, hiss). All three princes want to be the next king but in order to ascend to the now vacant Throne of Fire one brother has to destroy the other two. Which is where you come in.

The game offers one or two-player combinations, with the computer controlling either one or two princes depending upon the number of human players. Each prince begins with a force of nine men-at-arms, loyal followers who are under his control, scattered around the citadel so that you control a total of ten characters in all. In addition there is a fourth force, the King's Guard, who are neutral until one of the princes gains the Throne, at which point they throw their lot in with him, giving him a military advantage. There is also a number of Gate Rooms, where new men-at-arms can enter the castle. These new men become loyal to whichever prince's forces control the rooms, so it's to your advantage to try and secure one or more of these rooms.

Once you've selected your control options, number of players etc, the screen display is divided in half — one half for each prince (while the third just wanders around making a nuisance of himself). The top of each part of the display shows the room that the prince or one of his men occupies, and the lower part is a map of that part of the citadel, with the windows of each room coloured too indicate which force holds it.

Although you control ten or more characters in all, you can only control one at a time, so much of the skill of playing the game lies in co-ordinating these characters' actions so that you can explore the castle and secure the best locations.

Control of the game is divided amongst three modes of action. Select mode allows you to move a cursor around the citadel map in order to select which of your characters you want to control. Movement mode shows the animated movements of your character from room to room, and combat mode lets you swash a few buckles via joystick or keyboard control. The combat is swordplay, though there are a number of different weapons and magic objects that can be found around the place.

Most games players these days fall into one of two camps; there are the adventure/strategists, and the arcade zap fans. Throne of Fire however, falls rather uneasily between these two stools. The strategic element in the game is nowhere near as challenging or as wide in scope as Mike Singleton's earlier Lords of Midnight series, presumably because of the memory taken up by animated graphics. On the other hand, the combat sequences, though quite well animated, simply can't compete with dedicated arcade games.

The game is quite slickly programmed and enjoyable to play for a while, but I didn't find it all that addictive and getting to the end didn't really seem like a challenge that I wanted to spend much time on.

It might be that Throne of Fire was an attempt to combine the strategic complexities of Lords of Midnight with the more visceral thrills of games like Exploding Fist, and while that might be possible on larger machines I don't think you can pull it off with just 48K to play with.







Bugs in the ROM

Toni Baker details some of the Spectrum's 'undocumented features'

The Spectrum, as you may well know, is a machine of very many bugs. Some of these bugs crop up only in BASIC, some occur only in use of machine code, while others still are manifest all the time. Though the ROM (or rather ROMs, as there is now more than one type of Spectrum) cannot be written to, and therefore the bugs cannot in general be cured, knowledge of their existence can certainly help us to avoid them, or get round them in some way.

This article is therefore a compilation of all the bugs that I know of which occur in the 16K and 48K Spectrums, the 128 and 128+2, and also the ZX Interface One which runs the microdrives. Since there are many other interfaces which can be used in place of Interface One I cannot comment upon all of them. I shall restrict my focus of attention to the Spectrum plus Interface One, since these are Sinclair standard equipment.



It is almost impossible to classify bugs in any sensible manner. What I have done is to group them into fairly large and ambiguous clumps such as 'PRINT errors', 'CALCULATOR errors', and so forth. I personally have discovered only a minority of these bugs (some of which have caused me great frustration in the past), but the remainder are taken from two books, namely *Understanding Your Spectrum* by Dr Ian Logan, and *Master Your ZX Microdrive* by Andrew Pennell. Credit is given alongside all bugs other than those I have discovered myself. Doubtless other people will have made simultaneous discovery of these bugs — this is always the way.

I shall start off with three bugs which I couldn't fit into any category at all. The first bug is the most serious of all Spectrum bugs, for it is FATAL! This is the CLOSE # bug.

The Close # Bug
(from Understanding Your Spectrum by Martin Wren-Hilton)

The CLOSE # bug is present on all Spectrums *unless* a ZX Interface One is attached. If you attempt to close a stream which has not previously been opened using the BASIC CLOSE statement then the system will almost invariably crash. More often than not causing a system reset. The presence of the ZX Interface One cures this bug by transferring control to its own ROM (the Shadow ROM) whenever the program counter reaches address 1708h. The bug is caused by the table in the 48K ROM at address 1716 not being terminated by a 00 byte. It is interesting to note that Sinclair (or Amstrad rather) could have cured this bug at least in the Spectrum 128 — after all, they have made a few other alterations to the ROM — but alas they chose to leave the



bug in place, a surprising decision considering the seriousness of the bug.

The Plot Bug
The BASIC PLOT statement suffers from the fact that it will accept negative coordinates. In fact the statement PLOT X,Y really turns out to mean PLOT ABS(X),ABS(Y). This is caused by a failure to test the D and E registers on return from the CALL STK__TO__BC instruction at address 22DC.

The Screen One Bug
The Spectrum 128 offers a choice of two different areas of memory which may be used to hold the television picture. These are called SCREEN ZERO (the normal one, at address 4000h) and SCREEN ONE (the alternative, at address 7C000). The instruction **POKE 23388,24** will activate screen one, whereas **POKE 23388,16** will activate screen zero. (For more information see the series *Into the 128*, starting this month). The bug occurs in the rare circumstances of the silicon disc area becoming full (in which case files may collide with the screen) or if more than 217 files, however small, are SAVE!d (in which case the catalogue collides with the screen. The bug is caused by the complete failure of the RAMdisc system to recognise the existence of screen one. The BASIC program in Figure One will demonstrate this bug visually — it's quite fun to watch too.

PRINT Errors

The CHR$ 9 error
(from Understanding Your Spectrum by Dr Ian Logan)
CHR$ 9 is supposed to print as "forwardspace" (the opposite of backspace). In other words the effect of PRINT CHR$ 9 is *supposed* to be that the print position is moved rightward by one character square without altering the contents of the screen. Sadly, this does not work. Not only does the print position not move at all, but the character at the current print position is accidentally recoloured with the current colours. The first of these faults is due to the fact that the cursor right subroutine at address OA3D ends with RET, instead of with JP OADC,PO__STORE, however the second fault is due to sloppy thinking as the system variable MASK__T should have been assigned with FF and then restored. Personally I feel that the whole algorithm is up the spout.

The CHR$ 8 error
(from Understanding Your Spectrum by Dr Frank O'Hara)
CHR$ 8 prints as "backspace" — that is — when printed it will move the print position leftward by one character square. However - if the current print position is AT 1,0; then backspace will not work. Furthermore, it is possible to backspace from position AT 0,0; (which should not be possible) and this can produce some interesting results. The bug is caused by the cursor-left subroutine at address OA23 testing for line one instead of line zero. The byte at address OA33 should be 19h instead of 18h.

The STR$ error
(from Understanding Your Spectrum by Tony Stratton)
This is a bug which manifests itself in both BASIC and machine code. In BASIC the statement **PRINT "BUG"+STR$ 0.5** will simply print "0.5". In machine code the bug manifests itself in the PRINT. FP subroutine at address 2DE3, whose purpose is to print the floating point number at the top of the calculator stack to the current channel. This subroutine is called by the STR$ routine and in this way the bug also appears in BASIC. The but occurs whenever the number at the top of the calculator stack (or the argument of STR$) lies between −1 and +1 exclusive (excluding zero). In such a case an erroneous zero is left at the top of the calculator stack, and it is this which causes all the problems. In BASIC the bug can be avoided by using temporary



variables — eg **LET A$=STR$ 0.5: PRINT "BUG"+A$** will work. In machine code I would advise you to use the subroutine listed in Figure Two in place of PRINT__FP, as this takes the bug into account.

The Colour Commands error
(from Master Your ZX Microdrive, by Andrew Pennell)
If any channel other than the screen or the keyboard is selected as the current channel (ie if the last thing PRINTed was to a channel other than "K" or "S") then the permanent colour commands will erroneously transmit the colour control codes to the last channel used. The bug is caused by the failure to select channel "S" at the start of the subroutine CO__TEMP__1 at address 21E1. It is possible to get round the problem by preceeding each permanent colour command by a PRINT statement (eg **PRINT ;: INK 4**).

The Colour Controls error
If any non-standard channel (eg a user-defined channel) is selcted as the current channel then the colour statements such as PAPER 4 (etc) will produce *report C; Nonsense in BASIC* if the channel output subroutine returns with the carry flag set. (This caused me a few problems





Bugs in the ROM

Figure 1.

```
10 POKE 23388,24: REM activate screen one

20 FOR I = 1 TO 562

30 SAVE! "F" + STR$ I CODE 0,1

40 NEXT I

50 PAUSE 0

60 POKE 23388,16: REM activate screen zero
```

in the Streams and Channels series). This bug is only visible in machine code. To cure the problem it is necessary to ensure that all channel output routines return with the carry reset, at least for control codes 10h to 15h and their second parameter.

The RS232 Colour Commands error
This is for users of 128K machines only. On both the Spectrum 128 and the Plus2 the command **LPRINT INK 4** will produce error report *C; Nonsense in BASIC.* There are two reasons for this. Firstly, the new channel "P" (the built in RS232 socket) erroneously expects the colour controls to be followed by *two* parameters instead of just one, and secondly the ROM routine sets the carry flag (oops! see error above). Sinclair/Amstrad may like to know that the byte at 0086D (128) or 0088C (128+2) should be 01 instead of 02, and, further, that the byte at 0087C (128) or 0089B (128+2) which currently reads CCF should be replaced by two bytes SCF/CCF.

Figure 2.

```
2A655C    PRINT_FP_OK   LD HL,(STKEND)       HL: pts to end of calculator stack.
225F5C                  LD (X_PTR),HL        Store in system variable.
CDE32D                  CALL 2DE3,PRINT_FP   Print the floating point number.
2A5F5C                  LD HL,(X_PTR)        HL: pts to end of calc stack +5.
11FBFF                  LD DE,FFFB           DE:= -5.
19                      ADD HL,DE            HL: pts to end of calculator stack.
22655C                  LD (STKEND),HL       Restore (STKEND) to its correct value.
FD362600                LD (X_PTR)hi,00      Cancel error pointer.
C9                      RET                  Return.
```

Calculator Errors

The Division Error
(from Understanding Your Spectrum, by Dr Frank O'Hara.)
The BASIC statement **IF 1/2 <> 0.5 THEN PRINT "BUG"** is sufficient to show that there is a bug somewhere in the works. The but is in the division routine, which fails to register the thirty-fourth bit of a division. The JR displacement at address 3200h directs control to the label DIV_START instead of DIV_34TH. This means that numbers may not be correctly rounded up. You can avoid any major problems by using **IF ABS (X−Y < 2.15E+9** instead of **IF X=Y.**

The −65536 error
(from Understanding Your Spectrum, by Dr Ian Logan.)
The ROM is inconsistent about the calculator five-byte form of the number −65536. Some parts of the ROM assume that only numbers in the range−65535 to +65535 may be treated as integers, whereas other parts of the ROM allow an integer from of −65536 (as opposed to a full loading point form). This but is dealt with in detail in Dr Logan's book *The Complete Spectrum ROM Disassembly,* and is demonstrable by the now famous statment **PRINT INT −65536** which gives −1.

The SCREEN$ error
(from Understanding Your Spectrum, by Stephen Kelly and others.)
The BASIC routine for calculating SCREEN$ accidently leaves a duplicate entry of the string result at the top of the calculator stack. This means that the BASIC statement **IF "X"=SCREEN$ (0,0) THEN PRINT "BUG"** will invariably print "BUG" no matter what happens to be on the screen at the time. The bug is fortunately not present in machine code, and calling the subroutine S_SCRN$_S will evaluate SCREEN$ correctly leaving the stack correct. As with the STR$ errort the best way to avoid it in BASIC is to use temporary variables, eg. **LET A$=SCREEN$ (0,0): IF "X"=A$ THEN PRINT "BUG"** will work correctly.

The MOD.DIV error
Calculate code 32h (initiated by RST 28) is supposed to remove two items from the calculator stack — X and Y, say, and replace them by the quantitites X MOD Y, X DIV Y in that order. (Note: X MOD Y means X − Y * INT (X/Y), whereas X DIV Y means INT (X/Y).)

Unfortunately, it fails to take into account the fact that the INT subroutine corrupts calculator memory zero whenever its argument is negative. This means that the quantity X MOD Y will be incorrect whenever X/Y is negative. The but could have been cured had Sinclair ensured that the MOD,DIV subroutine at address 36AO utilised memory one instead of memory zero.

The E-TO-FP error
Calculator code 3C (initiated by RST 28) is supposed to multiply the item at the top of the calculator stack by the factor 10↑A. Unfortunately, unlike the B register, the A register is not preserved by RST 28, and so by the time the byte 3C is reached the A register will have already been corrupted. The only way out of the problem is to use endcalc/LD A,XX/CAL 2D4F, E_TO_FP/RST 28 instead of the byte 3C in your code.

The INKEY$#0 error
Normally, stream zero represents the keyboard, so we would expect INKEY$#0 to be similar in operation to INKEY$ (without a stream number). This, however, is not the case, and INKEY$#0 almost invariably produces an empty string, making it completely useless. Note also that calculator code 1A evaluates INKEY$#X (where X is the item at the top of the

calculator stack) and this too will be useless whenever stream X represents the keyboard. The but is in the subroutine at address 1634h which makes channel "K" the current channel. The instruction RES 5,(FLAGS) at address 1638h erroneously cancels out any key press which would otherwise have been detected. The bug could have been cured in the READ_IN subroutine at address 3645h by preserving the value of (FLAGS) during the call of CHAN_OPEN.

The FN error

When a user-defined function is evaluated the value of the system variable (CH_ADD) which stores the address of the next character to be interpretted, is saved on the machind stack instead of in one of the dynamic variables such as (X_PTR) during the evaluation of the function. This means that any user-defined function which causes the BASIC program area to move up or down in memory will cause error report *C; Nonsense* in BASIC. This is not normally possible, but exceptions can occur if the DEF FN statement contains the function USR, and could also conceivably happen with the function INKEY$#X. To avoid this bug you must ensure that any machine code subroutine which is called by a DEF FN statement does not disturb the BASIC program area (eg. it is impossible to define a function FN D(X,Y) which would delete BASIC lines X to Y inclusive without also manipulating the machine stack).

The USR error

Machine code programmers will be so familiar with this one that they don't even think of it as a bug any more. The problem is that the HL' register, whose value is required by the RST 28 sequence at address 2756h, is not preserved by the USR subroutine at address 34B3. This means that any machine code subroutine which disturbs the value of HL' (it should contain 2758h) will crash the system upon return. To avoid the bug simply insure that HL' contains 2758h upon execution of the final RET statement of any machine code subroutine called by USR.

Editing Errors

The Scroll? error

(from Understanding Your Spectrum, by Dr Ian Logan.)
Whenever a prompt message such as "scroll?" or the cassette messages appear, the Spectrum waits for you to press a key. The problem is that if you press the wrong key, things go wrong. Pressing TRUE VIDEO, INV VIDEO, CAPS LOCK, GRAPHIC or EXTEND

Figure 3.

```
FDE5        INT_ROUTINE   PUSH IY          Preserve the IY register.
FD213A5C                  LD IY,5C3A       IY: contains value required by
                                           normal interrupt routine.
FF                        RST 38           Execute normal interrupt routine.
FDE1                      POP IY           Restore IY register.
C9                        RET              Return.
```

MODE causes the previous edit line to appear at the bottom of the screen, with the 48K flashing cursor (even on 128K machines). This is because the KEY_INPUT subroutine at address 10A8 is designed to deal with CAPS LOCK, mode changing, and colour control parameters, however this is inappropriate for prompt messages.

The Current Line Cursor error

(from Understanding Your Spectrum, by Paul Harrison.)
This bug is applicable only on the 16K and 48K Spectrums. 128s don't have this problem because the EDIT key doesn't do quite the same thing. Type 9000 PRINT/9001/EDIT (note that "/" stands for ENTER) and, provided there are no lines greater than 9000, you'll see the problem — a GREATER-THAN symbol appears in the edit line. The bug occurs in the OUT_LINE routine at address 1855h. The bug could have been avoided if the sub-routine had refused to print the GREATER-THAN cursor whenever bit four of (FLAGS2) were set.

The Leading Space error

(from Understanding Your Spectrum, by Dr Ian Logan.)
Bit zero of (FLAGS) is supposed to be set whenever a leading space is not required for a keyword token. Unfortunately, the ROM is not consistent about its use, as the one-line program

CLS: FOR I=1 to 5: PRINT CHR$ 244: NEXT I

will prove. The problem would be solved if the flag were to be set every time CLS were executed, or whenever a control character 00 to 1F were printed.

The K-Mode error

(from Understanding Your Spectrum, by Chris Thornton.)
This is another one for 16K and 48K Spectrums only. If, when editing program line, the cursor is a flashing "K" then the editor is said to be in K-mode. This means that the next key you press will be interpretted as a keyword, so that if you press "P" for instance, then you'll get **PRINT**. The problem is that if you hold a key down so that it repeats then K-mode remains in force. This means that you could end up with a line like "NEXT NEXT" instead of "NEXT N". The bug is caused by the K_REPEAT routine at address 0310.

To work properly the routine would have to subtract hex A5 from the key value whenever this was greater than hex E5. This would replace keywords with capital letters.

Figure 4.

```
                          ORG 5B00
C3????      SWAP          JP SWAP_OK       Redirect control to revised routine.
                          ORG ????
F5          SWAP_OK       PUSH AF
C5                        PUSH BC          Stack regs A,B,C and flags.
ED5F                      LD A,R           P/V flag:= interrupt status.
F5                        PUSH AF          Stack interrupt status.
01FD7F                    LD BC,7FFD       BC:= port number reqd for paging.
3A5C5B                    LD A,(BANK_M)    A:= current pages.
EE10                      XOR 10           Complement 'ROM' bit.
F3                        DI               Disable interrupts (in case an
                                           interrupt occurs between the next
                                           two instructions.
325C5B                    LD (BANK_M),A    Store revised pages.
ED79                      OUT (C),A        Actually change ROM.
F1                        POP AF           P/V flag:= former interrupt status.
E2????                    JP PO,SWAP_OK_2  Jump if interrupts previously
                                           disabled.
FB                        EI               Re-enable interrupts.
C1          SWAP_OK_2     POP BC
F1                        POP AF           Restore A,B,C and flags.
C9                        RET              Return.
```

Bugs in the ROM



The Syntax Check error

(from Master Your ZX Microdrive, by Andrew Pennell.)
On the 16K and 48K Spectrums the four keywords ERASE, MOVE, FORMAT and CAT have incorrect syntaxes. The statements ERASE (string): MOVE (string),(string): FORMAT (string): CAT will be accepted as program lines, but will give error messages if you try to run them. On the 128K Spectrums the statement MOVE (string),(string) will be accepted as a program line, but treated as a REM statement on running. The bug in the 48K machine is due to the fact that the syntaxes of these Interface One commands were incorrectly anticipated. On the 128 a wholly new syntax routine is included, and the commands ERASE and CAT have been taken over for use with the silicon disc system, whilst FORMAT has been taken over for use with the RS232. MOVE, however, is not used at all, and it surprises me that the original false syntax has been reproduced in an entirely new syntax routine in the new ROM. Why?

Hook Code Errors

Hook codes are used to call subroutines in the Interface One's Shadow ROM. There are two errors in these:

The READ-N error

(from Master Your ZX Microdrive.)
Hook code 2F (READ_N) was intended to read a block of data from the "N" channel (the local area Network). The carry flag is assigned to indicate success or failure, but unfortunately the carry flag is subsequently destroyed by a call to the border restore routine.

The SET-T-MCH error

(from Master Your ZX Microdrive.)
On version one issues of the Interface One, hook code 2B (SET_T_MCH) accidently calls the OPEN_M routine instead of SET_T_MCH, making it identical in operation to hook code 22h. This bug has been cured in later edition of the Interface One, and in these later editions it correctly performs its task of creating a temporary "M" channel in the channel information area.

Interrupt Errors

The monopolising of IY

Though not strictly an "error" as such it is worth noting that the Spectrum's interrupt routine contains the instruction INC (IY+40), which is intended to increment the high byte of the (FRAMES) variable (when neccessary), however to do this IY must have a value of 5C3A at all times while interrupts are enabled (unless of course you define your own interrupt routines). Unfortunately, IY is not temporarily assigned with this value within the interrupt routine itself. The effect of this misdemeanour is that the value contained by IY must never be changed (unless interrupts are disabled or an alternative interrupt routine supplied). This restricts us, because it means that a machine code register which could have been available to us now must not be touched. It is possible to cure this condition by using the program of Figure Three as an interrupt routine.

The SWAP error

The SWAP routine which switches between the two ROMs of the 128K Spectrum invariably enables interrupts, which may not always be desired. Fortunately, this error may be cured by the program of Figure Four. Simply write the instruction JP SWAP_OK to location 5B00, where SWAP_OK is the address of the Figure Four program.

The PLAY error

At address 009CD in the 128, or 009EC in the 128+2, the STK_FETCH subroutine is called from the old ROM. At this point interrupts are disabled and IY is now being used as a pointer to the master PLAY information block. Unfortunately, because of the two bugs listed above, interrupts are enabled during the STK_FETCH call, leaving IY containing the wrong value. This means that if an interrupt were to occur during execution of the subroutine then ther would be a one in 65536 chance that (IY+40) will be corrupted — this corresponds to the volume setting for music channel A. In practice this circumstance is very rare and I have not known it to happen. Rewriting the SWAP routine as in Figure Four will cure this bug.

ERROR Errors

This last group of errors are bugs in the Spectrum's own error handling routines! Personally I find these most amusing:

The INVALID FILENAME error

(from Master Your ZX Microdrive.)
If the Interface One is attached then the error message "Invalid filename" is replaced by the message "Nonsense in BASIC", which is not particularly helpful.

The BREAK into CLOSE # error

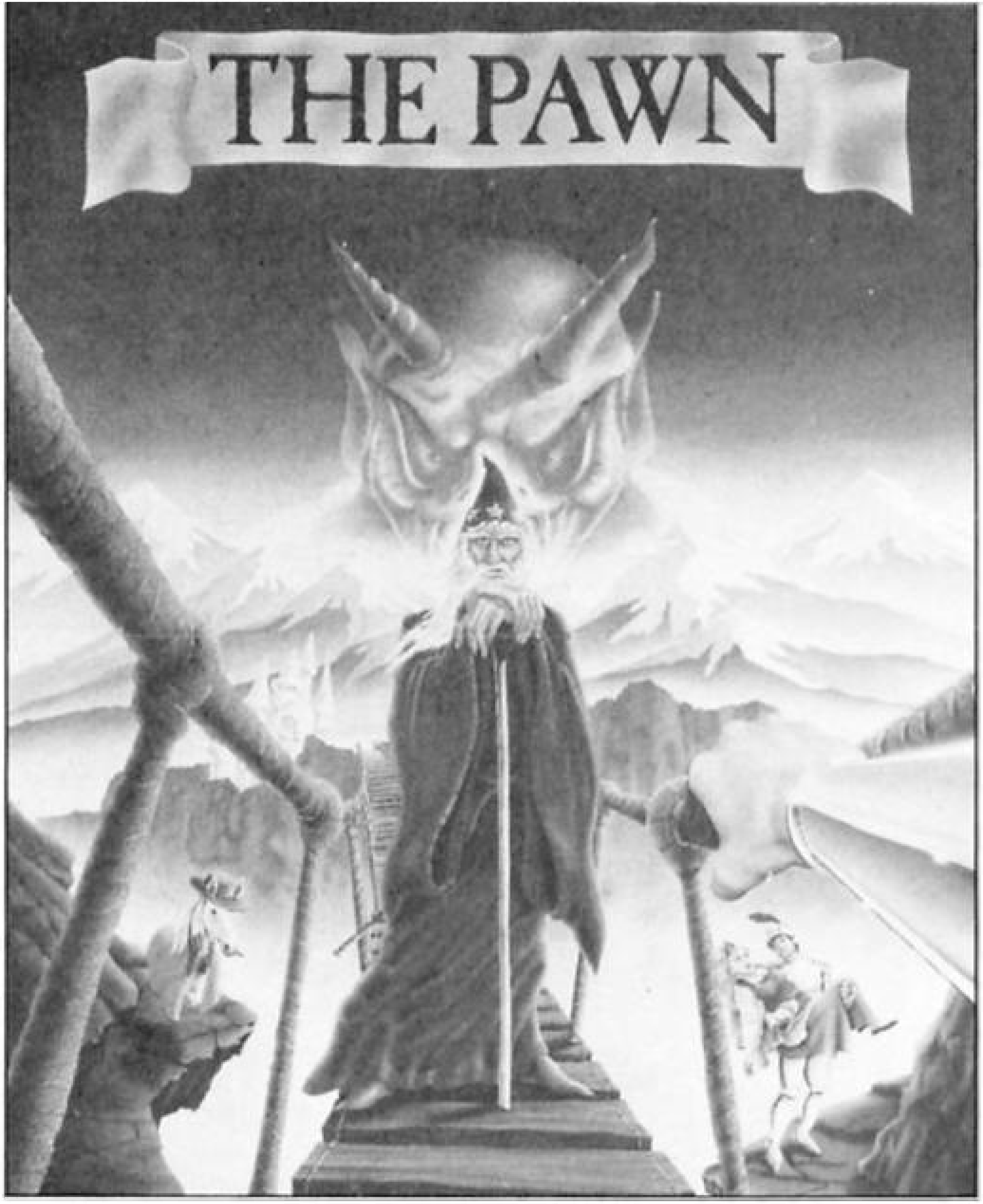
(from Master your ZX Microdrive.)
In version one of the Interface One it is possible to create unusable areas of memory which are not reclaimable (except in machine code of course). If you BREAK into a **CLOSE** statement which refers to one of the Interface One channels then the memory used by the stream is not reclaimed. The bug has been cured for version two of the Interface One.

The INVALID DEVICE EXPRESSION error

Most ironic of all is the curious error which only occurs with the combination of ZX Interface One and Spectrum 128 or 128+2. If you have a BASIC line such as

1000 OPEN 4,"X"

(the line number must be greater than 999) then you should by rights get the error message "Invalid device expression, 1000:1" — indeed, true to form, the Spectrum does indeed *attempt* to produce this message for us. Unfortunately, the message is just too long to fit on a line! The thirty three characters in the error message will not fit on a line of only thirty two squares. The Spectrum 128's marvellous little ful screen editor only allows for a lower screen of two lines (including one blank line above the error message), but his particular error message takes three lines, not two! If you're using only the lower two lines as the editor then the effects of the but aren't too bad: you merely get a few spurious things happening at the bottom of the screen. If you're using the whole screen as an editor heover then the bug is FATAL, causing an unavoidable system reset in 48K mode! A shame really — the poor thing was only trying to tell you about a bug in your BASIC. Never mind — these things happen!



Gordon Hamlott takes his 128 into the mystical land of Kerovnia.

The Pawn (128K only)
Rainbird
£14.95

Owners of 128K Spectrums (Spectra?) must be beginning to wonder if software houses actually take them seriously. The best games that they have had so far are reworkings of old 48K standards, perhaps with some extra rooms or better music to keep them happy.

Fear not! All that has now changed with the release of *The Pawn* — a superb adventure written by Magnetic Scrolls and released by Rainbird. Even so, you are not getting all the game I am afraid, as even with all the extra memory available, there is still no room for the pictures that made this game famous on machines like the ST and Amiga.

The Pawn has had a somewhat strange development. Originally written as a text only game for the QL. It was then instantly forgotten about, like most QL software. It next appeared on the Atari ST complete with some stunning graphics — the best ever seen in an adventure by several orders of magnitude. Conversions were then written for most disk based systems as the illustrations could be loaded in separately where necessary. But there is simply not enough room in the 128, so it's back to text only.

This is not necessarily a cause for gnashing of teeth and tearing of hair, for even in this form, the game knocks spots off any other Spectrum adventure.

Kerovnia

The game is set in the magical land of Kerovnia. The old King, Erik, is losing his grip on the country. There have been assassination plots and rumours of insider trading within the Farthington Real Ale Company. This is an attempt to stop a take over by the Roobikyoub Dwarfs who are distillers of the finest whisky around. On top of all that, there is a general election in the offing with one of the main candidates being Gringo Baconburger, a dwarf violently opposed to the royal family. Into the middle of this political turmoil comes you.

Your first problem is that you are wearing a wrist band which, try as you may, you cannot remove. This is unfortunate since it prevents you from crossing a thin red line that surrounds Kerovnia. You haven't got the slighest idea at this stage what on earth is going on or what you are supposed to be trying to do. As you travel about, so you get a message to deliver, meet a guru who falls about laughing whenever he sees you, find a snowman standing guard outside a tower, some alchemists arguing about whose turn it is to get the supper and of course, you have the chance to vote.

Location descriptions are long and vivid, so much so that the lack of illustrations doesn't seem to matter too much. The problems are both both original and difficult — don't expect to complete this one in one sitting.

The main thing that sets The Pawn apart from other adventures is its parser, which can quite justifiably claim to be state of the art — even better than the legendary Infocom parser. This may seem like a small point to pay so much attention to but it soon becomes apparent that playing the game is a lot easier than most adventures. Not because the problems are easy but because you can enter commands as you would say them naturally. There is no need to go searching for exactly the right combination of words.

One example taken from the instruction booklet, and not I hasten, to add part of the game, shows how powerful commands can be strung together. "Get all except the cases but not the violin case then kill the man eating shrew with the contents of the violin case. Remove the shrew's tail and use it to tie the pole and the noose together."

Pretty impressive stuff even if the command is a load of gibberish.

The game comes beautifully packaged — a box containing two booklets, a map and a 64 page novella written with the tongue never once leaving the cheek. This serves two purposes. Firstly as an anti-piracy device, as you need to look up certain words and secondly as a help section. A number of clues are available to you but only if you are prepared to type in a lengthy cipher. Even then, you may not be allowed access to the clue if you have not solved enough of the game and the answers that you get can be downright misleading at times. You have been warned.

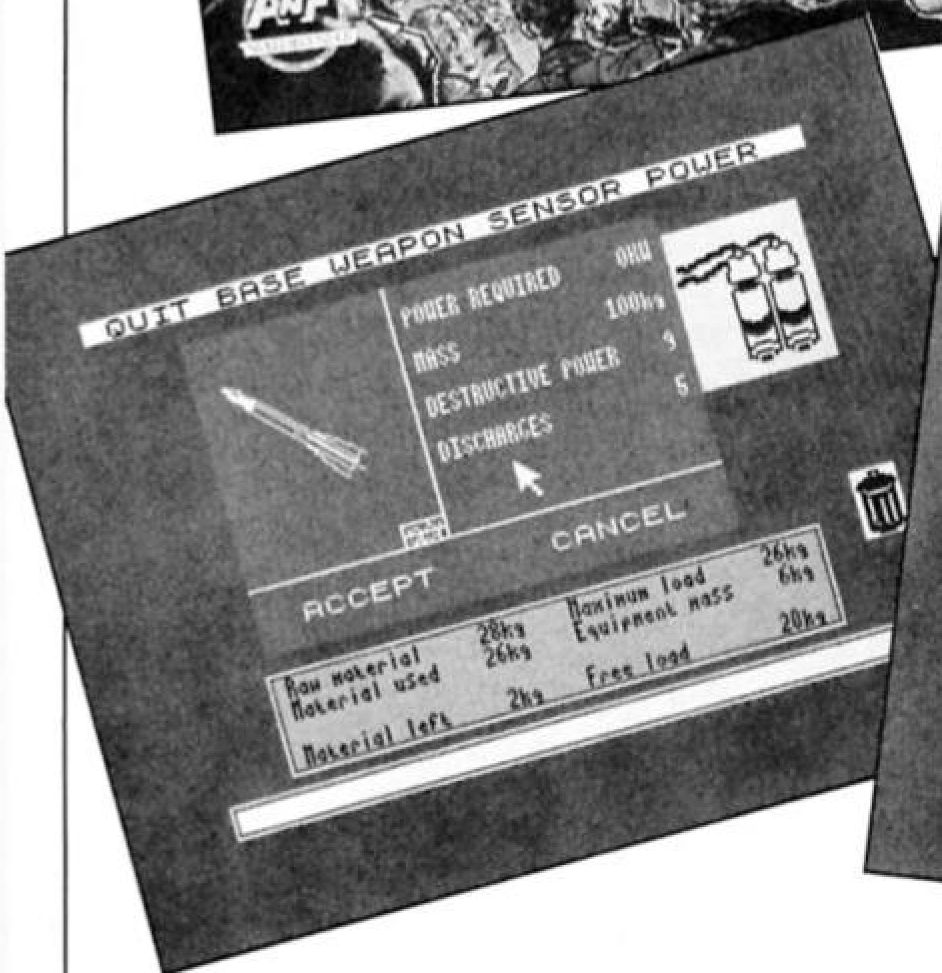
It's a pity about the pictures but you can't have everything! Perhaps if Alan Sugar does get round to launching his Spectrum disk system they will eventually materialise but don't let that thought stop you buying a copy of the game now. If you enjoy adventures, then The Pawn is a must for your collection. Even if you are not a fan, buy a copy and see what state of the art adventuring is all about.

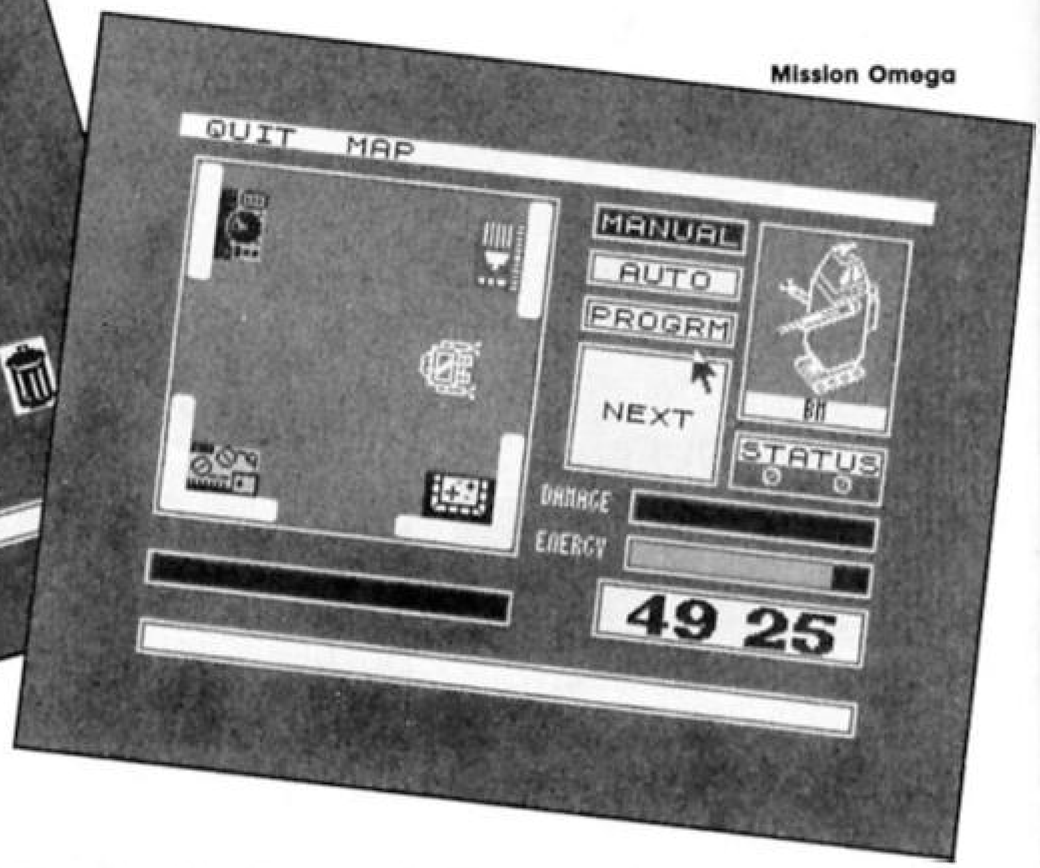




SPACE GAMES OFFER

Three space games from Argus Press Software for just £12.95





Mission Omega

In the third special offer arranged in conjunction with Argus Press Software, ZX readers can buy three very different space games at a discount price.

Xeno is a futuristic sports simulation played out in an oval arena. Your objective is to slam the puck over the polished playing surface into your opponents goal. It's a fast and furious game that can be played by two players or against the computer.

Mission Omega is an action strategy game in which earth is under. This icon driven game sets you the task of building and then co-ordinating a force of robots to combat the invaders.

DefCom is a space game in the shoot 'em up tradition of wholesale destruction. Taking the "Star Wars" technology as a starting point, Defcom simulates attacks on earth to be countered by your mastery of the satellite defence systems.

These three games are available at a price of £12.95 including p&p only by mail order. Do not send your cheques or postal orders to the ZX address but to the address on the coupon printed on this page. Make cheques and postal orders payable to Argus Press Software. Please allow 28 days for delivery.

ARGUS PRESS SPACE GAMES OFFER

Please send me Xeno, Mission Omega and DefCom.

Name ....

Address ....

....

....

I enclose a cheque/postal order for £12.95 including p&p payable to Argus Press Software.

Send your remittance to Argus Mail Order(ZX), Units 1 & 2, Conlon Developments, Watery Lane, Darwen, Lancashire BB3 2ET.

Are you decisive? Could you command a space freighter? Do you hire the Glork or Fligronk?

PSI-5 TRADING COMPANY

PSI-5 Trading Company
US Gold
£9.95

Commanding a star freighter for the Psi-5 Trading Company isn't as easy as it's portrayed in games like Elite. The problems lie in the fact that you have to carry perishable cargo to far flung parts of the galaxy, in a scrap heap of a ship that's crewed by a collection of galactic oddballs, while being shot at by every pirate in the sector.

Having selected the easiest option (nothing's easy in this business!) of carrying 'nucleric' 120 parsecs to Kozzar-7, it's time to interview 30 candidates for your five crew positions. For each position you have six weirdos to choose from and can even punch up their record, but I doubt that'll help you. Since trial and error will sort out the good from the bad and the ugly, I opted for a 200 year old hairy Glork to handle the weapons, a man-hating Palagonan to scan space for enemy ships, a yellow blob called Yenx to navigate, the green scaly Fligronk to man the engineering department and finally T3XR9 to repair things (since he's a robot I thought he'd know what to fix and when). With the crew in position and me in the captain's chair, we headed for the stars.

The screen display is a mass of windows, instruments and displays as you issue orders to your crew that show your exterior view, speed, compass, damage and supply gauges, a picture of the crew member you're talking to and their controls.

You can select the crew and then their actions from duck-shoot menus and order engineering to get power through to the standard systems, navigation to plot a course and accelerate, scanning to search and identify other ships and weapons and repair to standby. That's just to get started!

The fun and games really start when another ship approaches. If it's a friendly ship and you accidentally fire at it you could break a long standing treaty but if it's an enemy you can't afford to leave it alone. Now you must act quickly as scanning must get a fix before weapons can fire any of it's four weapons either at a specific target or at will. The battle is shown in your view window but you probably won't get a chance to watch it as things really heat up. Messages flood in from your crew as systems are damaged and need repairing.



If you survive you may get a chance to organise repairs before the pirates strike again. Fail, and you'll be destroyed and the pirates will ransack your ship. Either way your mission report will count up the profits or losses of your mission as it deducts repairs costs, casualties and lost allies from cargo profits, early bonuses and bounty for zapping the Chilank, Zeltoad, Flarkan and Trantot pirates.

There's no doubt that Psi-5 Trading Company is an excellent game but is anyone good enough to play it?





SPECTRUM GAME REVIEW



Head Over Heels
Ocean
£7.95

The authors of Batman return with another hit.

Once upon a time Ultimate released a game called Knight Lore which had real 3D graphics, a cute hero and lots of traps and things. And lo, the sky parted and millions of games with real 3D graphics, cute heros etc, rained down upon the firmament until the firmament was crawling with the blasted things and the arrival of each new game was greeted with yawns and "Oh look, another Ultimate style 3D game."

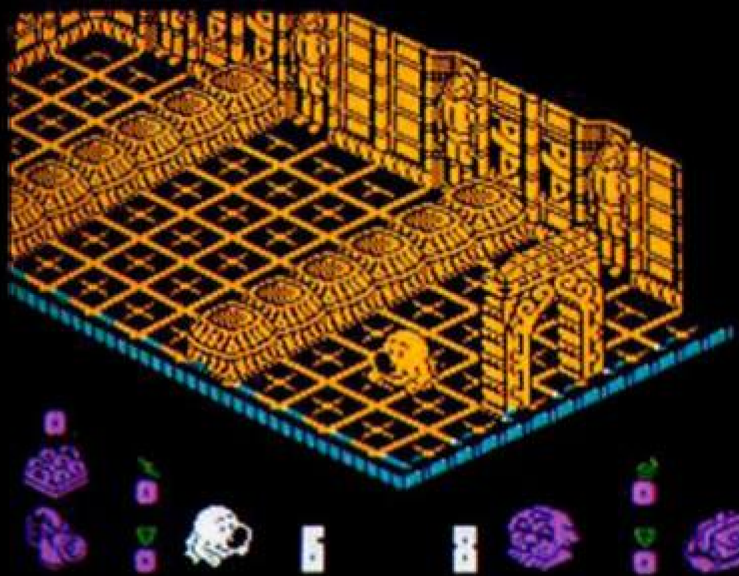
Most of these games had the look of Ultimate's games, but didn't match the 'feel', the combination of tricky problem solving and playability. Then Ocean released their Batman game, which had some character of its own and stood out among all the other Ultimate clones. Now, the Batman authors, John Ritman and Bernie Drummond, have produced Head Over Heels and have improved the format even further.

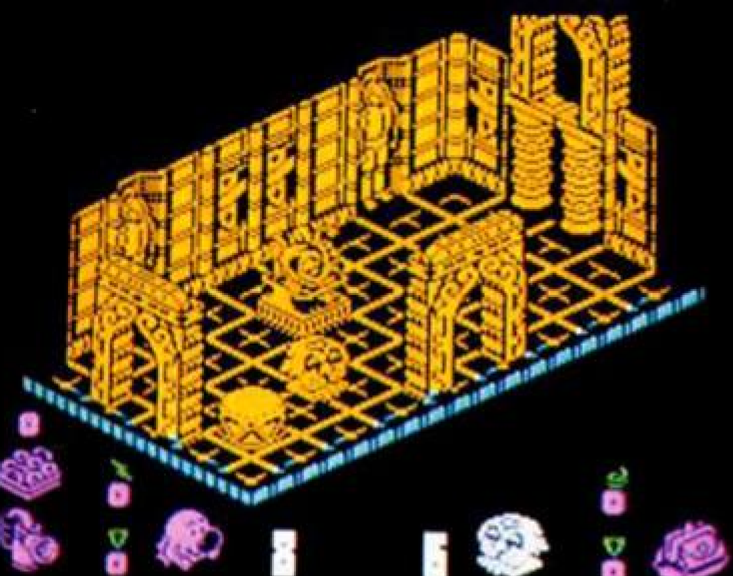
The plot of the game concerns the Blacktooth Empire, a group of four planets enslaved by the rulers of the planet Blacktooth. Your task is to recapture the four crowns, one hidden away on each planet, in order to cause an uprising, and finally to get the citizens of Blacktooth itself to revolt against their masters. Boiled down, this means 'wander around and find the objects', but it's the execution not the plot that makes it so enjoyable.

Brought to Heel

You are put in control of two beings called Head and Heels, who have developed a symbiotic relationship allowing them to exist seperately or to join together and combine their abilities. Head is a winged reptile with the ability to glide, but who is completely legless (in a literal sense), while Heels has developed powerful legs but seems to have misplaced his arms.

At the start of the game Head and Heels have been captured, separated, and imprisoned in different parts of Castle Blacktooth. Graphically the game looks just as you'd expect — rooms with various moving objects, monsters, and problems to solve — but it's with the two





characters you control that the game departs from the usual format. Your first task is to get both Head and Heels out of the castle so that they can be reunited. To do this the program allows you to switch control from one character to the other. This might not sound revolutionary but it's an original touch that adds a whole new element to the game. Each character has to find his way out of the castle by a different route, and has his own supply of extra lives, objects to collect, and so on. This makes it like playing two games at once, yet you also have to organise things so that Head and Heels can still get together for the later stages of the game. It's no good getting Head out of the castle with all his lives left (upto ten available) if Heels is left trapped somewhere with just one life left.

Sometimes their paths almost cross and you can find Head and Heels in the same place, but they can't quite touch each other because of some sort of barrier, so you have to send them off in different directions again.

In addition, it takes a while to figure out quite what each character is capable of. Head's gliding abilities can come in very useful, but it takes a bit of experimenting to discover how best to control him. There are some tasks, such as climbing ladders, that took me ages to get the hang of. Heels is easier to control as he just hops about a little bit, but being the strong one he carries most of the important objects.

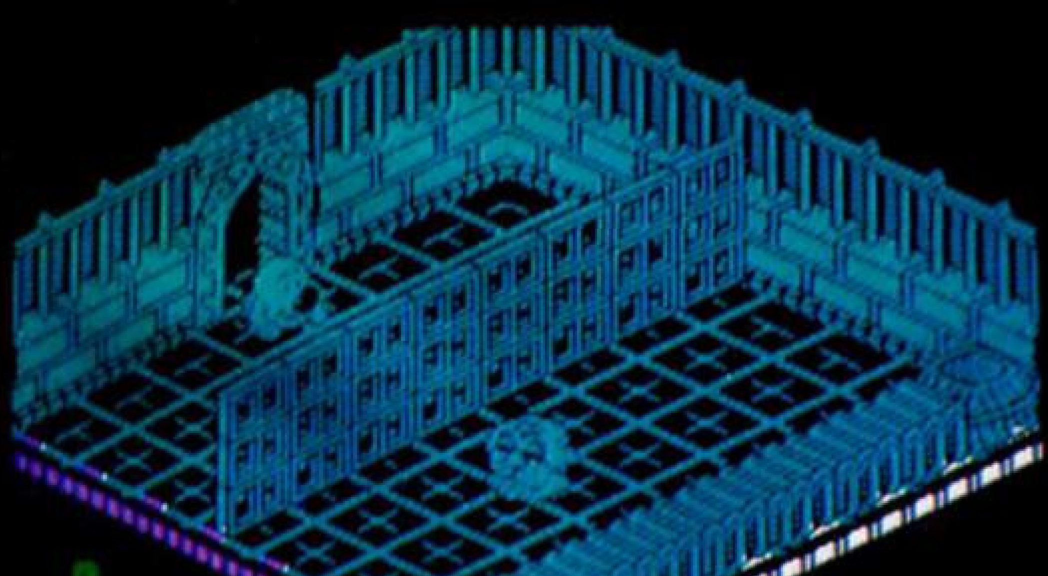
Reincarnation Fish

Scattered around the various locations are such exotic items as the Reincarnation Fish (a sort of save game option which allows you to restore the game to the stage where you first found the fish), the deadly donuts, hush puppies, cuddly stuffed rabbits, teleports, and all the weird paraphenalia that you'd expect in this sort of game.

Once you've reunited the dynamic duo your problems still aren't over as you've got the four planets, comprising around 300 screens, to explore. The problems that lie in wait for you are all cleverly thought out — hard enough to take a bit of thought, but not so hard that they're discouraging. What makes solving all these traps and obstacles so intriguing is the fact that Head and Heels, once united, can still split up and function separately. This means that whenever you're faced with a problem you have to work out whether it can be solved by Head or Heels as individuals, or the two of them together.

I've played other games where control is divided between two or more characters, but these have always seemed rather fiddly and cumbersome to play, but here the game has been thought out so that the choice of which character(s) to control is actually part of the challenge and adds to the game.

I'd thought that this style of game had more or less been milked dry, but Head Over Heels proved an unexpected surprise, and one that will keep you occupied for hours.



MONSTER HIT

INTO THE 128

Part Two by Toni Baker

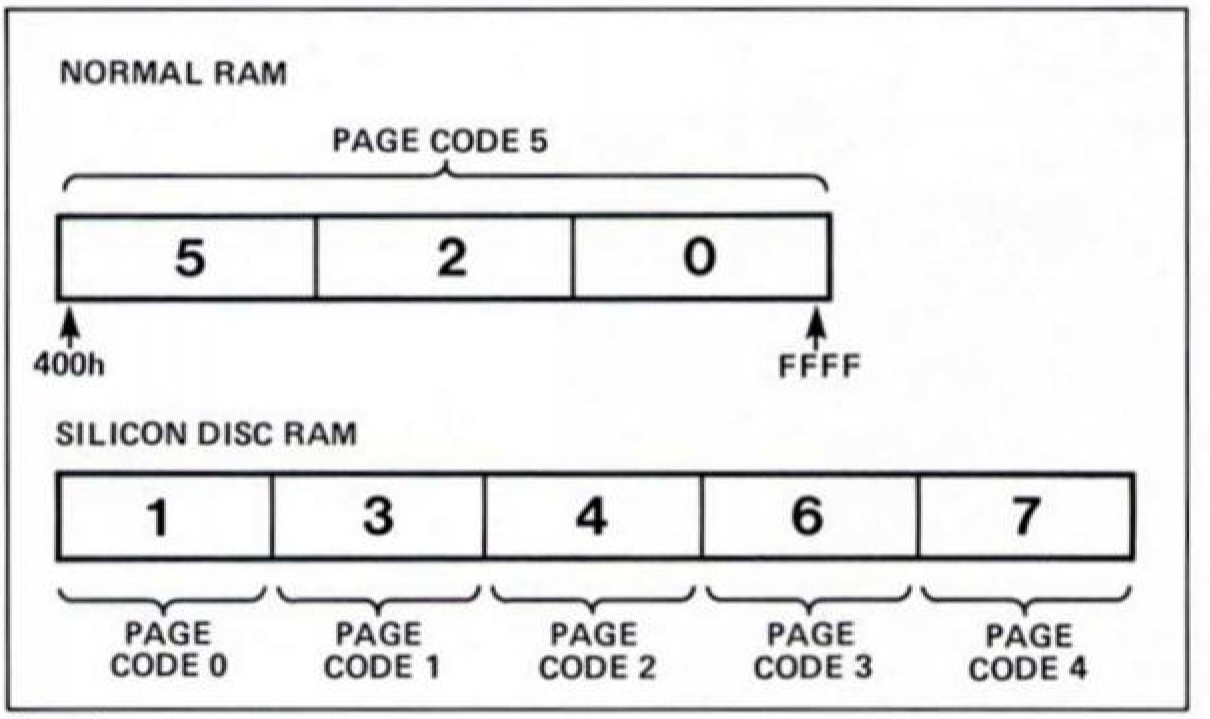


Figure 1.

In this, the second part in our series on the Spectrum 128 and the +2, we turn our attention to the file storage system called the SILICON DISC, or RAMDISC. In BASIC it is possible to save programs, data, or code onto silicon disc in much the same way that you can save them onto cassette, except that the command for accessing the silicon disc system is **SAVE!** instead of just **SAVE.** Silicon disc has the advantage of speed — the file is saved seemingly in an instant.

Likewise there are corresponding commands **LOAD!, VERIFY!** and **MERGE!** The *disadvantage* of silicon disc is that everything stored there will be wiped out when the power is disconnected.

In this article we are going to look at the silicon disc organisation from a machine code point of view. In point of fact, the terms "silicon disc" and "RAMdisc" are not very accurate. There is no disc of any description inside the 128; there is only 128K of RAM memory. The system is called "silicon disc" purely because it *mimics* a normal disc drive, but a disc system it certainly is not.

RAMpage

Last month I told you about ROM and RAM pages. This month, I must dwell on the subject of RAM pages for a little longer. As you recall there are eight RAM pages numbered from 0 to 7, and each page contains 16K, making 128K in all. What you now need to realise is that these RAM pages are divided into two distinct groups, which we may refer to as "NORMAL RAM" and "SILICON DISC RAM". Figure One shows this distinction diagramatically. As you can see, normal RAM contains three 16K pages, making 48K in all — in this sense the 128 is not really much different from the old 48K Spectrums. Normal RAM contains a screen, some system variables, the current BASIC program together with its BASIC variables and channel information, the calculator stack, the user-defined graphics, and so on — just as before. Everything *new* in the 128 happens in the *other* section of RAM — the silicon disc area.

Normal RAM consists of RAM pages 5, 2 and 0 (in that order). You will recall that page 5 is permanently mapped to address 4000h, and that page 2 is permanently mapped to address 8000h. Under normal circumstances you will find that page zero is paged in and mapped to address C000 — this means that normal RAM contains 48K of memory with continuous addressing from 4000h all the way up to FFFF.

The same cannot be said, however, for the silicon disc area. This consists of RAM pages 1, 3, 4, 6 and 7 (in that order), however *none* of these pages are permanently mapped anywhere. This means that in order to access *part* of the silicon disc memory you must page in one of its pages, which will then reside at address C000 in place of RAM page zero (which must be restored afterwards). It is not possible to access all of the silicon disc area at once.

To make life easier, I shall introduce the concept of PAGE CODES. You see, each of the eight RAM pages possesses a PAGE CODE, which is a number between zero and five.

NB: THE PAGE CODE IS NOT THE SAME THING AS THE PAGE NUMBER.

It is the use of page codes which enables us to distinguish between the two different regions of RAM, and to access the silicon disc area sensibly. Pages 5, 2 and 0 (ie normal RAM) all have a page-code of five. Pages 1, 3, 4, 6 and 7 (ie the silicon disc area) all have page-codes less than five — in fact they run in sequence: 0, 1, 2, 3, 4.

This makes addressing of the Spectrum's memory a little simpler. Last month we established the convention whereby every byte in RAM could be referred to by a five digit hexadecimal number — for instance, the first byte in the silicon disc area would be referred to as address 1C000. The "1" at the start refers to RAM page one, and the remaining four digits are the address within that page.

```
INC HL
BIT 7,H
RET NZ
LD HL, C000
INC A
RET
```

Figure Two

We can do almost exactly the same thing with page codes. For instance, consider again the first byte in the silicon disc area. This exists on RAM page one, but page one has a paging-code of zero (see Figure One). Therefore, we can specify the byte uniquely by supplying the page code (zero), and the address (C000). To avoid confusion with absolute page numbers, we will *not* fuse these together into a single five digit number. We will, instead, define a new convention as follows: place the page-code within BRACKETS, and follow it by the address.

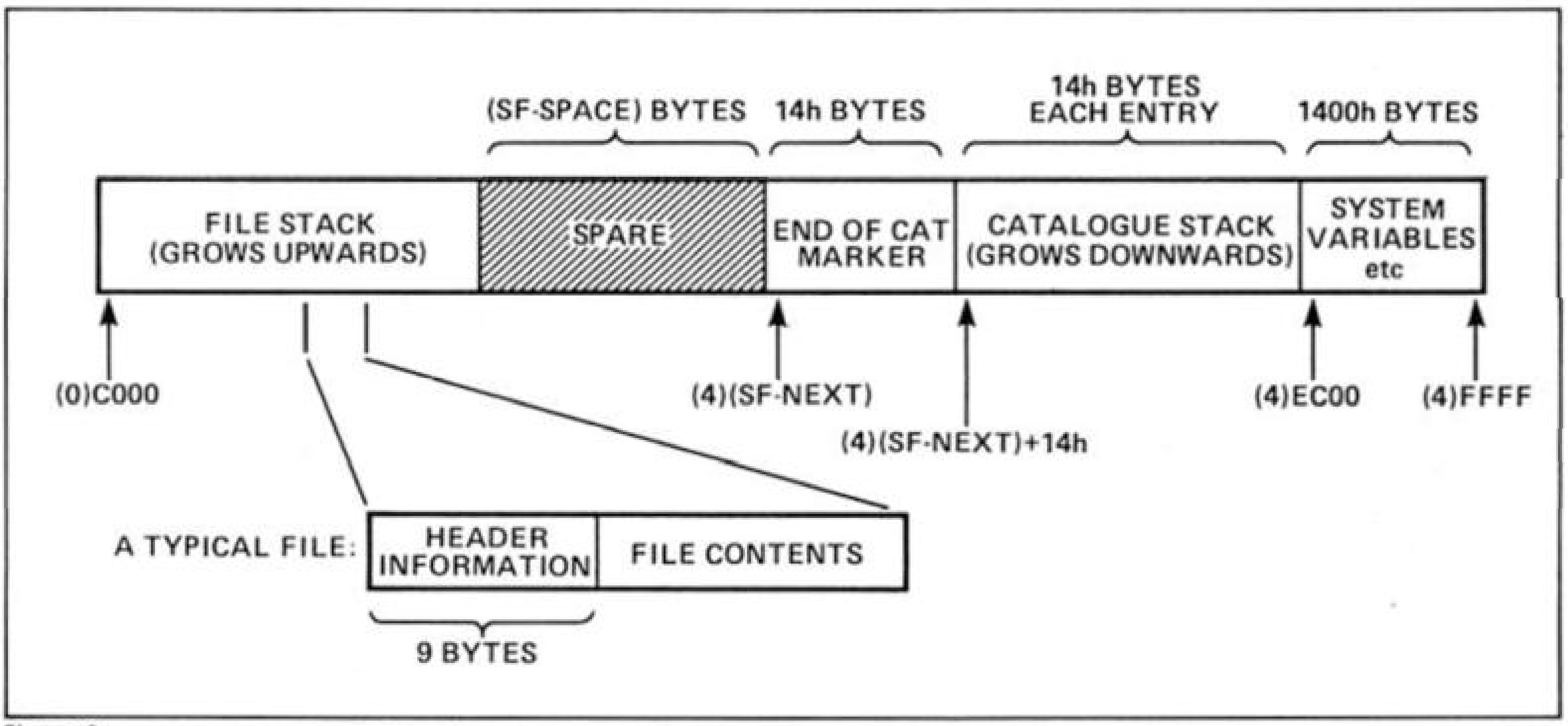


Figure 3.

Using this second convention, the first byte of RAMdisc, in addition to having an absolute address of 1C000, also has a PAGE-CODED ADDRESS OF (0)C000.

Just to give you a better grasp of the new convention, here are some equivalents, in both absolute and page-coded address systems:

```
1FFFF = (0)FFF
3C001 = (1)C001
4D800 = (2)D800
6EE00 = (3)EE00
7EBEC = (4)EBEC
 4000 = (5)4000
 ABCD = (5)ABCD
 F000 = (5)F000
```

(Note that in the old absolute convention the last three items could also be referred to as 5C000, 2EBCD and 0F000 respectively).

I hope all this makes sense. The point is that a single register pair will not hold a 128K address, since a register pair can only hold four hex digits. Therefore, to hold such an address in machine code we will require *three* registers, not just two. We may use either the absolute or the page-coded convention, whichever is most convenient at the time. You will find that when working with the silicon disc area, page-coded addresses are a lot more helpful. As an example, suppose we wanted to store the page-coded address (4)EBEC in the register triplet AHL. To do this A must contain 04, whilst HL must contain EBEC. As long as you remember that this is a page-coded address and not an absolute address then you won't go far wrong.

The advantage of page-codes is that they make everything nice — for instance, the whole of normal RAM has the same page-code (5), and furthermore, the page-codes of the silicon disc area run sequentially: (0), (1), (2), (3) and (4). We may use this fact to our advantage in machine code. For instance, suppose AHL contains the page coded address of a byte in the silicon disc area. How may we calculate the page-coded address of the byte by following it? Figure Two shows a possible solution. Imagine how messy this simple subroutine would become if we were to use absolute addresses instead!

The first nine bytes of a standard file contain the following header information:

```
HD_00 (one byte):  File type. 00=program
                              01=numeric array
                              02=string array
                              03=bytes of code
HD_0B (two bytes): Length of file, excluding header info.
HD_0D (two bytes): Start address from which file was SAVE!d.
HD_0F (two bytes): Length of program, or name of array.
HD_11 (two bytes): Auto-run line number.
```

Figure Four

Memory

The very first subroutine in this month's main program is called PAGE_(A), and its purpose is to page in the RAM page whose page-code is held in the A register. If the page code specified is (5) then it will page in RAM page zero, restoring normal RAM. The subroutine works by converting the page-code to an absolute page number and then paging in normally. This subroutine is the key to using silicon disc memory, and both the 128 and the 128+2 contain similar sub-routines in their new ROM.

Now it is time to look closer still at the silicon disc memory. Figure Three shows how this memory is organised. Growing upwards from address (0)C000 is the file stack. Every time a file is saved it is added to the top of the file-stack. Each entry in the file-stack contains nine bytes of header information followed by the file itself as it would be saved on cassette or microdrive. Figure Four shows the meanings of the nine bytes of header information wich precede each program, DATA or CODE file. Note that under normal circumstances this header is transferred to the system variables HD_00 to HD_11 (5B71 to 5B79) while a file is being processed.

Growing downwards from address (4)EBFF is the catalogue stack. Each entry takes exactly twenty bytes, so the first entry will begin at address 4(EBEC), the second entry at (4)EBD8, and so on. Each entry in the catalogue stack is a reference to one of the files in the file stack — in other words, there are always exactly the same number of entries in the catalogue stack as there are files in the file stack. The top of the file stack is pointed to by the system varible (SF_NEXT) at address 5B83. This is a two byte system variable containing an address which is *always* assumed to point into page seven (ie page-code (4)). This means that the catalogue stack can never grow beyond (4)C000 — as a consequence there is a maximum limit on the number of entries the catalogue stack may contain — five hundred and sixty-two in fact. This means that

the file-stack may in turn contain *at most* 562 files, and therefore the maximum number of files which may be saved in RAMdisc is 562, however small the files might be.

At the top of the catalogue stack is a twenty byte information block called the "end-of-cat" marker. Only three of its bytes are important, however. Figure Five shows the information contained by each entry in the catalogue stack, whilst Figure Six shows the corresponding information contained by the end-of-catalogue marker.

Bearing all of this in mind, we can now examine the main program of this article, which is called SUPERCAT! The program may be called from BASIC with the command: **RANDOMIZE USR 33200** which runs the machine code from the label SUPERCAT. Note that (RAMTOP) *must* be less than C000, so that the switching of pages will not affect the machine stack.

The program will list on the screen a complete and highly detailed catalogue of everything saved in RAMdisc — it will tell you what kind of file it is, its auto-run line number (if it has one), its intended location (if it's CODE); and so on. If you examine this program you should see how the silicon disc area works, quite comprehensively. If you're a masochist you might also like to refer to the last article in the STREAMS AND CHANNELS series which ended quite recently in ZXC.

Next month I'll be looking at the possibilities for extending BASIC on the 128. See you then.

This is a typical entry from the catalogue stack, assuming that IX points to the first byte:

IX+00	SF_NAME	(ten bytes):	Filename, with trailing spaces if necessary.
IX+0A	SF_START	(three bytes):	Page-coded address of start of file in file-stack.
IX+0D	SF_LEN	(three bytes):	Length of file, including header info.
IX+10	SF_END	(three bytes):	Page-coded address of first byte in file-stack beyond end file.
IX+13	SF_FLAG	(one byte):	Normally reset (unless file construction incomplete).

Figure Five

This is the information contained by the end-of-catalogue marker, assuming that IX points to the first byte. Note that this marker is indexed by (4)(SF_NEXT).

IX+00		(ten bytes):	Not used.
IX+0A	SF_START	(three bytes):	Page-coded address of first spare byte in RAMdisc.
IX+0D		(seven bytes):	Not used.

Figure Six

```
                        ORG 8000h
C5         PAGE_(A)     PUSH BC
FE05                    CP 05
2002                    JR NZ,PAGE_(A)_2        Jump unless normal RAM reqd.
3EFF                    LD A,FF                 A will contain 0 after increment.
3C         PAGE_(A)_2   INC A
47                      LD B,A                  B:= page_code + 1.
CB38                    SRL B                   Divide B by two.
80                      ADD A,B                 A:= absolute RAM page number.
F610                    OR 10                   Include 'ROM 1' bit.
01FD7F                  LD BC,7FFD              BC:= port no reqd for paging.
325C5B                  LD (BANK_M),A           Store as current pages.
ED79                    OUT (C),A               Page in required RAM page.
C1                      POP BC
C9                      RET

                        ORG 8018
50524FC7   PROG_TEXT    DEFM PROG               Text to be printed by program.
444154C1   DATA_TEXT    DEFM DATA
3F3F3FBF   QUES_TEXT    DEFM ????
4C494EC5   LINE_TEXT    DEFM LINE
0D425954   USED_TEXT
45532055
5345C4                  DEFM BYTES USED
2046494C   LEFT_TEXT
45534259
54455320
4C4546D4                DEFM FILESBYTES LEFT
17170028   MAX_TEXT
35363220
4D4158A9                DEFM (562 MAX)

                        ORG 804F
1A         MESSAGE      LD A,(DE)
E67F                    AND 7F                  A:= char to print (bit 7 ignored).
D7                      RST 10                  Print character.
1A                      LD A,(DE)
13                      INC DE                  DE: points to next chr.
17                      RLA                     Carry:= end of string bit.
30F7                    JR NC,MESSAGE           Loop if more to come.
3E20                    LD A,"space"
D7                      RST 10                  Print a trailing space.
C9                      RET                     Return.
```

```
                        ORG 805C
7C         HEX_HL       LD A,H                  A:= high part of HL.
CD6180                  CALL 8061,HEX_A         Print in hex.
7D                      LD A,L                  A:= low part of HL.
F5         HEX_A        PUSH AF                 Stack byte to print in hex.
1F                      RRA
1F                      RRA
1F                      RRA
1F                      RRA                     Low nibble:= first hex digit.
CD6A80                  CALL 806A,HEX_A_L       Print first hex digit.
F1                      POP AF                  Low nibble:= last hex digit.
E60F       HEX_A_L      AND 0F                  Isolate low nibble.
F630                    OR 30                   Digits "0" to "9" correct.
FE3A                    CP 3A
3802                    JR C,HEX_A_L_2          Jump with digits "0" to "9".
C607                    ADD A,07                Digits "A" to "F" also correct.
D7         HEX_A_L_2    RST 10                  Print digit.
C9                      RET                     Return.

                        ORG 8076
AF         DECIMAL_HL   XOR A                   A:= 00.
E5         DECIMAL_AHL  PUSH HL
D5                      PUSH DE                 Stack registers.
C5                      PUSH BC
F5                      PUSH AF                 Stack high byte of AHL.
44                      LD B,H
4D                      LD C,L                  BC:= low word of AHL.
CD2B2D                  CALL 2D2B,STACK_BC      Push onto calculator stack.
F1                      POP AF                  A:= high byte of AHL.
CD282D                  CALL 2D28,STACK_A       Push onto calculator stack.
EF                      RST 28                  Switch calculator on.
                                                Lo,Hi
34004100                stk data 65536
                        (91 00 00 00 00)        Lo,Hi,65536
04                      multiply                Lo,65536*Hi
0F                      add                     X
2E                      str$                    STR$ X
38                      end calc                Switch calculator off.
CDF12B                  CALL 2BF1,STK_FETCH     BC:= length of string;
                                                DE:= address of string.
F1                      POP AF                  A:= required no. of chrs.
F5                      PUSH AF
91                      SUB C                   A:= reqd no of leading spaces.
```

```
3809                    JR C,DECIMAL_P          Jump if string too long.
2807                    JR Z,DECIMAL_P          Jump if string exactly right leng
6F                      LD L,A                  L:= reqd no of leading spaces.
3E20      DECIMAL_LOOP  LD A,"space"
D7                      RST 10                  Print leading space.
2D                      DEC L
20FA                    JR NZ,DECIMAL_LOOP      Loop to print remaining spaces.
CD3C20    DECIMAL_P     CALL 203C,PR_STRING     Print number in decimal.
C1                      POP BC
D1                      POP DE                  Restore registers.
E1                      POP HL
C9                      RET                     Return.

                        ORG 8045
060A      PRINT_FILE    LD B,0A                 B:= length of filename.
7E        PF_LOOP       LD A,(HL)               A:= next chr from filename.
23                      INC HL
D7                      RST 10                  Print character.
10FB                    DJNZ PF_LOOP            Print whole of filename.
3E20                    LD A,"space"
D7                      RST 10                  Follow it by a space.
5E                      LD E,(HL)
23                      INC HL
56                      LD D,(HL)               DE:= address of file.
23                      INC HL
7E                      LD A,(HL)               A:= page code of start of file.
23                      INC HL
4E                      LD C,(HL)
23                      INC HL
46                      LD B,(HL)               BC:= low word of length of file.
23                      INC HL
C5                      PUSH BC
4E                      LD C,(HL)               C:= high byte of length of file.
E1                      POP HL                  CHL:= length of file.
EB                      EX DE,HL                CDE:= length of file;
                                                AHL:= page coded address of file.
F5                      PUSH AF
E5                      PUSH HL                 Stack page-coded address of file.
2A7A5B                  LD HL,(SC_00)
3A7C5B                  LD A,(SC_00+2)          AHL:= no of bytes used so far.
19                      ADD HL,DE
89                      ADC A,C                 AHL:= new subtotal.
D5                      PUSH DE
111400                  LD DE,0014
19                      ADD HL,DE
8A                      ADC A,D                 Include 20d bytes for cat entry.
D1                      POP DE
227A5B                  LD (SC_00),HL
327C5B                  LD (SC_00+2),A          Store new subtotal.
E1                      POP HL
F1                      POP AF                  AHL:= page coded addr of file.
C5                      PUSH BC
D5                      PUSH DE                 Stack length of file.
010900                  LD BC,0009              BC:= length of header info.
11715B                  LD DE,HD_00             DE: points to header buffer.
F5        PF_HD_PAGE    PUSH AF
CD0080                  CALL 8000,PAGE_(A)      Page in next byte of header info.
F1                      POP AF
EDA0      PF_HD_BYTE    LDI                     Transfer one byte into buffer.
E2F280                  JP PO,PF_TYPE           Jump if all 9 bytes transferred.
CB7C                    BIT 7,H
20F7                    JR NZ,PF_HD_BYTE        Jump if next byte on same page.
2100C0                  LD HL,C000              HL: pts to start of (next) page.
3C                      INC A                   Increment coded page number.
18EC                    JR PF_HD_PAGE           Jump back for remaining bytes.

                        ORG 80F2
E1        PF_TYPE       POP HL
C1                      POP BC                  CHL:= length of file.
3A715B                  LD A,(HD_00)            A:= file type.
111880                  LD DE,PROG_TEXT
A7                      AND A
280D                    JR Z,PF_MESSAGE         Jump if file is PROGRAM.
1E1C                    LD E,DATA_TEXT lo
3D                      DEC A
2808                    JR Z,PF_MESSAGE         Jump if numeric array.
3D                      DEC A
2805                    JR Z,PF_MESSAGE         Jump if string array.
3D                      DEC A
2835                    JR Z,PF_CODE            Jump if CODE.
1E20                    LD E,QUES_TEXT lo
CD4F80    PF_MESSAGE    CALL 804F,MESSAGE       Print file type.
3A715B                  LD A,(HD_00)            A:= file type.
FE04                    CP 04
300F                    JR C,PF_STANDARD        Jump unless file is non-standard.
3E17                    LD A,17
D7                      RST 10
3E1A                    LD A,1A
D7                      RST 10
D7                      RST 10                  PRINT TAB 26;
79                      LD A,C                  AHL:= length of file.
0606                    LD B,06                 Signal "six digits required".
CD7780                  CALL 8077,DECIMAL_AHL   Print length of file.
185D                    JR PF_EXIT              Jump to exit.
A7        PF_STANDARD   AND A
2038                    JR NZ,PF_DATA           Jump unless file is PROGRAM.
217953                  LD HL,HD_11+1
CB7E                    BIT 7,(HL)
2046                    JR NZ,PF_LEN_EXIT       Jump unless auto-run line no
                                                exists.
112480                  LD DE,LINE_TEXT
CD4F80                  CALL 804F,MESSAGE       PRINT "LINE ";
2A785B                  LD HL,(HD_11)           HL:= auto-run line number.
0600                    LD B,00                 Signal "indefinate no of digits".
CD7680                  CALL 8076,DECIMAL_HL    Print auto-run line number.
1836                    JR PF_LEN_EXIT          Jump to exit.
2A745B    PF_CODE       LD HL,(HD_0D)           HL:= address of CODE block.
E5                      PUSH HL
CD5C80                  CALL 805C,HEX_HL        Print in hex.
3E2D                    LD A,"-"
D7                      RST 10                  PRINT "-";
ED5B725B                LD DE,(HD_0B)           HL:= length of CODE block.
19                      ADD HL,DE
2B                      DEC HL                  HL:= addr of last byte of block.
CD5C80                  CALL 805C,HEX_HL        Print in hex.
3E20                    LD A,"space"
D7                      RST 10                  Print a space.
E1                      POP HL                  HL:= address of CODE block.
0605                    LD B,05                 Signal "five digits".
CD7680                  CALL 8076,DECIMAL_HL    Print in decimal.
3E2C                    LD A,","
D7                      RST 10                  Print separating comma.
181A                    JR PF_CODE_EXIT         Jump to exit.
47        PF_DATA       LD B,A                  B:= file type.
3A765B                  LD A,(HD_CF)            A:= variable name.
E63F                    AND 3F
F640                    OR 40                   Convert to ASCII character.
D7                      RST 10                  Print it.
05                      DEC B
2803                    JR Z,PF_DATA_2          Jump if numeric array.
3E24                    LD A,"$"
D7                      RST 10                  Print "$";
3E28      PF_DATA_2     LD A,"("
D7                      RST 10                  Print "(";
3E29                    LD A,")"
D7                      RST 10                  Print ")";
3E1B      PF_LEN_EXIT   LD A,1B
CDC30A                  CALL 0AC3,TAB_A         PRINT TAB 27;
2A725B    PF_CODE_EXIT  LD HL,(HD_0B)           HL:= length of BASIC file.
0605                    LD B,05                 Signal "five digits".
CD7680                  CALL 8076,DECIMAL_HL    Print in decimal.
3E0D      PF_EXIT       LD A,0D
D7                      RST 10                  Print a newline.
3E04                    LD A,04
C30080                  JP 8000,PAGE_(A)        Page in catalogue and return.
                        ORG 8188
00000000  MINIMUM       DEFB 00,00,00,00
00000000                DEFB 00,00,00,00
0000                    DEFB 00,00

                        ORG 8192
FFFFFFFF  MAXIMUM       DEFB FF,FF,FF,FF
FFFFFFFF                DEFB FF,FF,FF,FF
FFFF                    DEFB FF,FF

                        ORG 819C
C5        CP_NAME       PUSH BC
D5                      PUSH DE
E5                      PUSH HL
EB                      EX DE,HL                DE: pts to name used for
                                                comparison.
60                      LD H,B
69                      LD L,C                  HL: points to filename
                                                under test.
060A                    LD B,0A                 B:= length of filename.
1A        CP_NM_LOOP    LD A,(DE)
13                      INC DE
BE                      CP (HL)                 Compare next character.
23                      INC HL
2002                    JR NZ,CP_NM_EXIT        Jump if characters different.
10F8                    DJNZ CP_NM_LOOP         Repeat comparison for all chrs.
E1        CP_NM_EXIT    POP HL
D1                      POP DE
C1                      POP BC
C9                      RET                     Return.
```

```
                       ORG 81B0
3E02      SUPERCAT     LD A,02
CD0116                 CALL 1601,CHAN_OPEN        Select stream two (upper screen).
217A5B                 LD HL,SC_00
0603                   LD B,03
3600      SC_RESET     LD (HL),00
23                     INC HL
10FB                   DJNZ SC_RESET              Reset 'number-of-bytes-used' and
                                                  'number-of-files' counters.

3E04                   LD A,04
CD0080                 CALL 8000,PAGE_(A)         Page in RAM page containing cat.
218881                 LD HL,MINIMUM
119281    SC_OUTER     LD DE,MAXIMUM
01ECEB                 LD BC,EBEC                 BC: points to first catalogue
                                                  entry.

E5        SC_INNER     PUSH HL
2A835B                 LD HL,(SF_NEXT)            HL: points to 'end-of-cat' marker.
A7                     AND A
ED42                   SBC HL,BC
E1                     POP HL
2818                   JR Z,SC_FOUND              Jump if whole of catalogue has
                                                  been searched.

CD9C81                 CALL 819C,CP_NAME
3009                   JR NC,SC_NEXT              Jump if name preceeds or equals
                                                  minimum, alphabetically.

EB                     EX DE,HL
CD9C81                 CALL 819C,CP_NAME
EB                     EX DE,HL
3802                   JR C,SC_NEXT               Jump if name follows maximum,
                                                  alphabetically.

50                     LD D,B
59                     LD E,C                     Set new upper limit.
E5        SC_NEXT      PUSH HL
21ECFF                 LD HL,FFEC
09                     ADD HL,BC
44                     LD B,H
4D                     LD C,L                     BC: points to next cat entry.
E1                     POP HL
18DE                   JR SC_INNER                Repeat for all cat entries.
E5        SC_FOUND     PUSH HL
219281                 LD HL,MAXIMUM
A7                     AND A
ED52                   SBC HL,DE
E1                     POP HL
2808                   JR Z,SC_DONE               Jump if all files dealt with.
EB                     EX DE,HL                   HL: points to catalogue entry.
E5                     PUSH HL
CDA580                 CALL 80A5,PRINT_FILE       Print info for this file.
E1                     POP HL
18C6                   JR SC_OUTER                Repeat for all files in cat.
112880    SC_DONE      LD DE,USED_TEXT
CD4F80                 CALL 804F,MESSAGE          PRINT "BYTES USED ";
2A7A5B                 LD HL,(SC_00)
3A7C5B                 LD A,(SC_00+2)             AHL:= number of bytes used.
0606                   LD B,06                    Signal "six digits".
CD7780                 CALL 8077,DECIMAL_AHL      Print number of bytes used.
3E17                   LD A,17
CDC30A                 CALL 0AC3,TAB_A            PRINT TAB 23;
21ECEB                 LD HL,EBEC
ED5B835B               LD DE,(SF_NEXT)
A7                     AND A
ED52                   SBC HL,DE                  HL:= length of cat stack - 14h.
01ECFF                 LD BC,FFEC                 BC:= -14h.
11FFFF                 LD DE,FFFF                 DE:= -1.
13        SC_FILES     INC DE                     DE: counts files.
09                     ADD HL,BC
38FC                   JR C,SC_FILES              Count all files.
EB                     EX DE,HL                   HL:= number of files.
0603                   LD B,03                    Signal "three digits".
CD7680                 CALL 8076,DECIMAL_HL       Print number of files.
113380                 LD DE,LEFT_TEXT
CD4F80                 CALL 804F,MESSAGE          PRINT "FILES/BYTES LEFT ";
2A855B                 LD HL,(SF_SPACE)
3A875B                 LD A,(SF_SPACE+2)          AHL:= number of bytes unused.
0606                   LD B,06                    Signal "six digits".
CD7780                 CALL 8077,DECIMAL_AHL      Print number of bytes remaining.
114380                 LD DE,MAX_TEXT
CD4F80                 CALL 804F,MESSAGE          PRINT TAB 23;"(562 MAX)"
3E0D                   LD A,0D
D7                     RST 10                     Print extra newline at end.
3E05                   LD A,05
C30080                 JP 8000,PAGE_(A)           Page in normal RAM and return.
```

Disc Manager
Bradway Software
£10.50

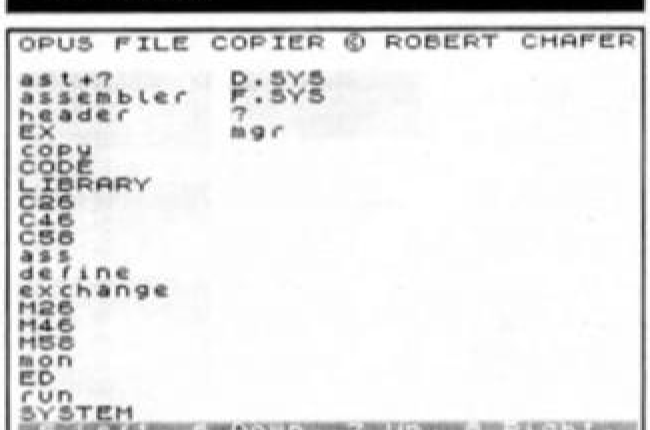


Figure 1. File copier menu.

Disc Manager from Bradway Software is a comprehensive file and disc maintenance utility, including a file copier, a disc doctor, and comprehensive 9-page manual. The loader program provides you with four options: Disc doctor, File transfer, Select load disc, Select save disc. Disc doctors provides facilities for examining the free space left on the disc, printing a full catalogue, or getting even more information about a file. It allows you to select files for erasing, rename a file or read a block from the disc into the Spectrum memory; the contents can then be examined, edited and finally written back to a disc.

The arrangement of the files on a disc, along with the location of the empty sectors is provided by the "list block" facilities which will also list to the printer. "Modify block" allows you to alter the contents of a block and the "write block" facility permits saving the contents of an edited block back on the disc after suitable checks.

The file copier allows you to select and copy a file or groups of files from one disc to another. The program checks to make sure that the save or load disc is in the appropriate drive. It copied a disc from my 178K 3.5" drive to my 720K 5.25" drive without difficulty. However, it does this by loading files into the computer memory (29K is available) and awaiting a keypress before saving: patience is necessary, or a large brick on the keyboard! The copy program is optimised to use the minimum disc space and files are copied in the order selected. This allows you to keep precious groups of files together, and also to assign a precedence to files which have to be saved on a fairly full disc.

Finally, the instruction manual contains some invaluable information on how to recover damaged or erased files (the erase, of course, merely deletes the catalogue title). It tells you how to search the disc, check the blocks, re-insert a name into the directory, and recover the file that went missing. Great stuff!

Most of my gripes concern two-disc systems or large capacity discs. For those with a single disc system, I would recommend this program; go ahead and buy it. It is still helpful to users of twin discs of similar sizes, and indeed, useful, though not perfect, for those who have discs of two different sizes. My principal irritations were with the clumsy transfer system; with a two disc system, the transfer, once initiated, should be automatic to the end. Similarly, you have to go right back to the start every time you wish to change source or destination drive, and then reload the transfer program; not a good idea. There is no single-key "compact" option, no "rename disc" option, and no single-key "format" option: this is useful if you wish to extend the catalogue file, particularly necessary with large capacity discs, but again this facility also is lacking.

Hopefully, for users of more complex disc systems, an upgrade is in the pipeline.

John Wase

The British Empire attempts to strike back.

Zulu War
CCS
£8.95



ZULU WAR



In 1879 General Chelmsford, commanding the British forces in Africa decided that the only way to stop the disputes between the Boers and the Zulus was to take over their two countries and govern them. The resulting "police" action was one of the worst massacres in British history as the highly trained British troops gunned down the Zulu warriors. This latest wargame from CCS has biased the action in the Zulu favour to produce a challenging war-game with a difference.

Two separate scenarios are included on either side of the game tape in which between 20,000 and 48,000 Zulus attack unprepared British regiments.

The British begin both scenarios arranged in their rigid square formations that you must manoeuvre between the broken walls and trees that form this barren landscape. The size of a square that is made up from soldier icons indicates the size and strength of the regiment and range from the strongest 5 x 5 down to a single soldier, although even this represents 60 trained soldiers. The Zulus swarm at you in their famous horned beast formation with the main body supported by two flanking horns. This tramples the enemy under a wave of spears. Even though the figures are little more than UDGs which shuffle along at a snail's pace, they make a refreshing change from the usual wargaming symbols and remind me of my old figure war-gaming days.

The game itself is a fascinating clash between two totally different but well organised armies. The British have the advantage of fire power and can gun down the Zulus from a distance and have the formidable lancer and cavalry regiments. These key soldiers can not only shoot from a distance but can also break Zulu lines with charges and fight ferociously with their swords. The Zulus outnumber the British by thousands and use this advantage effectively as they attack in massive waves. They are faster than the armed British and are formidable hand-to-hand fighters. As British commander you'd better not let them get too close or you'll quickly lose troops and firepower and face an Alamo style defeat.

Three levels of play ensure that there is a scenario that you can win, one where you just get beaten and one that's a total disaster. This forms a highly addictive formula that will drive you back for more.



GUNRUNNER

A super-smooth shoot 'em up from Hewson.



Gunrunner
Hewson
£7.95

Gunrunner has an instantly forgettable plot (save the planet from doom etc) but very memorable gameplay that will undoubtedly appeal to arcade players.

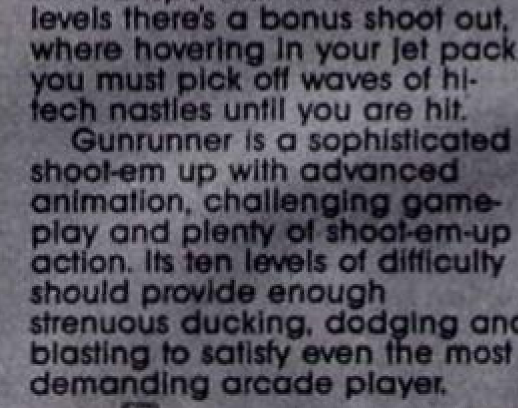
Even the basic gameplay idea sounds less than promising — it's a left to right scrolling game in which you manoeuvre your Gunrunner character over a pipe system, avoiding hovering scanning orbs, bombs, deadly domes and various lethal flying objects. The execution of the game, however, is superb, your character is remarkably mobile and very smoothly animated. When a string of flying gadgets appears behind you your character can duck down, turn and shoot in one fluid movement. But this is just one of his actions, he's just as good in the running and jumping departments too. All this agility is essential as the hazards appear very quickly and in no set order. If you go back to the beginning having lost one of your lives there's no telling how the defending hardware will have re-grouped for the next game.



Along the way there are objects to pick up extra fire-power, poison shields and a jet pac. The poison allows you to clear the screen of nasties several times, the shield allows temporary invulnerability againt direct hits and the jet pack will give you a short-lived airborne interlude.

If you manage to fight your way through a level you must look forward for a pedestal which will transport you to the next level. A good (but exasperating) touch occurs as you stand on the pedestal when a last minute wave of nasties can dispose of you before you are transported. In between levels there's a bonus shoot out, where hovering in your jet pack, you must pick off waves of hi-tech nasties until you are hit.

Gunrunner is a sophisticated shoot-em up with advanced animation, challenging game-play and plenty of shoot-em-up action. Its ten levels of difficulty should provide enough strenuous ducking, dodging and blasting to satisfy even the most demanding arcade player.



QL COLUMN

Brian Becket with news on the QL scene.

FATE is not kind and my QL went the way of all flesh the other week. Its back in working order but not soon enough for me to play around with some software packages long due for a review in these hallowed pages. So general news and rumour will have to do for this month. First of all, let me correct a previous report on ABC Elektronic's up-coming/soon-to be up-coming QL up-grade kit. Contrary to previous reports, it does not include disc drives or a disc interface but the main box does have room for their inclusion. Those cunning West German's at ABC may devise good QL products but their English leaves something to be desired and their handout a couple of Microfairs ago was worded in such a way that I read the inclusion of disc drives in the up-grade kit. So forget any great bargain and while the ABC kit still looks good, it isn't quite as good a deal as it looked back then. With a bit of luck, the folk at ABC will hire an English PR release writer and concentrate their time on other and better things.

In the meantime, Digital Precision (which is handling the ABC upgrade kit in this country) is releasing its own keyboard and an upgrade kit is available. I'm not sure at the moment whether the upgrade kit is the ABC one or not but the keyboard is Digital's. Its known as the "Deluxe Keyboard" and sells for £89.95. It has 83 keys with on-board number pad and is designed for maximum typing convenience. It looks suspiciously like a prototype keyboard I saw exhibited at the same price at a recent Micro-fair by some company or other based in the North and, if its the same one, it looks like a bit of wheeling and dealing has been going on and the Digital Empire is growing by leaps and bounds. The QL ship has gone under and somebody had to reach the lifeboat first. Given Digital's excellent product record I'm glad it was them and — if Digital acquires a virtual monopoly on QL products or their distribution, it should prove no bad thing.

Digital's deluxe keyboard is compatible with all QL disc drives/interfaces, RAM expansions, printers, mice and software. In other words, it is fully compatible with the myriad products and addions which are either still available or had their brief moment of glory in the marketplace. Its an external board so you get another peripheral but it connects easily by cable to an interface fitting into the 8049 co-processor socket. No soldering or electronics experience is required. There are five user-programmable function keys. Its black and matches the old Sinclair box that we known and love so well. I haven't had a chance to type on one yet but (if it is the same keyboard as the prototype I played with at that Microfair) the feel of the keys is excellent and rivals the BBC and my very own electronic typewriter. I will be reporting further details when I get one to try for real, kick around and otherwise abuse for your enlightenment.

Meanwhile, over at Schon, the company's slot-in QL keyboard is now available with plastic moulded keys and an external IBM-type QL board is due on the market in a few weeks. If Schon meets its schedule, it should be available by the time you read all this so give them a call and ask how much it costs (they neglected to tell me) or whatever else you want to know about it before I report further on the matter. Besides, Schon likes calls. The MD called me up a week or so ago to ask when my review of the slot-in keyboard was coming out because he hadn't had any calls from anybody out there as yet. I pointed out that his keyboard review was on the stands at the moment but didn't mention that this column just might not have any readers to call him up. So help prove that there is intelligent, QL-life in the universe and ring up Schon if you are interested in a keyboard or if you would like to show them that somebody out there really does care.

For the latest on the software front, we have to go back to Digital yet again. Soon to be released is a desk-top publishing package or (as the company was being a shade quiet on the subject for the moment) something along these lines. This can only be good news as desk-top publishing is not only all the rage but is ideally suited to the QL which has few other markets to appeal to. A month or so ago, I briefly reviewed the **Front Page** QL desk-top program for the QL, which so far has the field pretty much to itself, and noted the QL's growing popularity with small groups putting out news-sheets, handouts and so on. With the QL being cheap (if you can still find one on the High Street) or a second hand bargain, its a natural for groups without a lot of money to throw around who need a quality computer with lots of good, economical software available. The Front Page is a good product with, as the company admits, certain limitations dictated by the product's low price. If Digital is entering the field, I would expect an excellent product at a reasonable price.

Last month, I suggested that all diehard QL owners without disc drives immediately start looking for some. The last few weeks have only reinforced this opinion and, to risk beating an old drum, I stand behind this opinion even more firmly than before. I've now had several more microdrives go haywire and while most of the data can be saved the things are now as out-dated as the pyramids and far less interesting. If you are going to keep the old QL ticking over for any serious purpose, disc drives are inevitable and you might as well bite the bullet now as later.

BUDGET GAMES

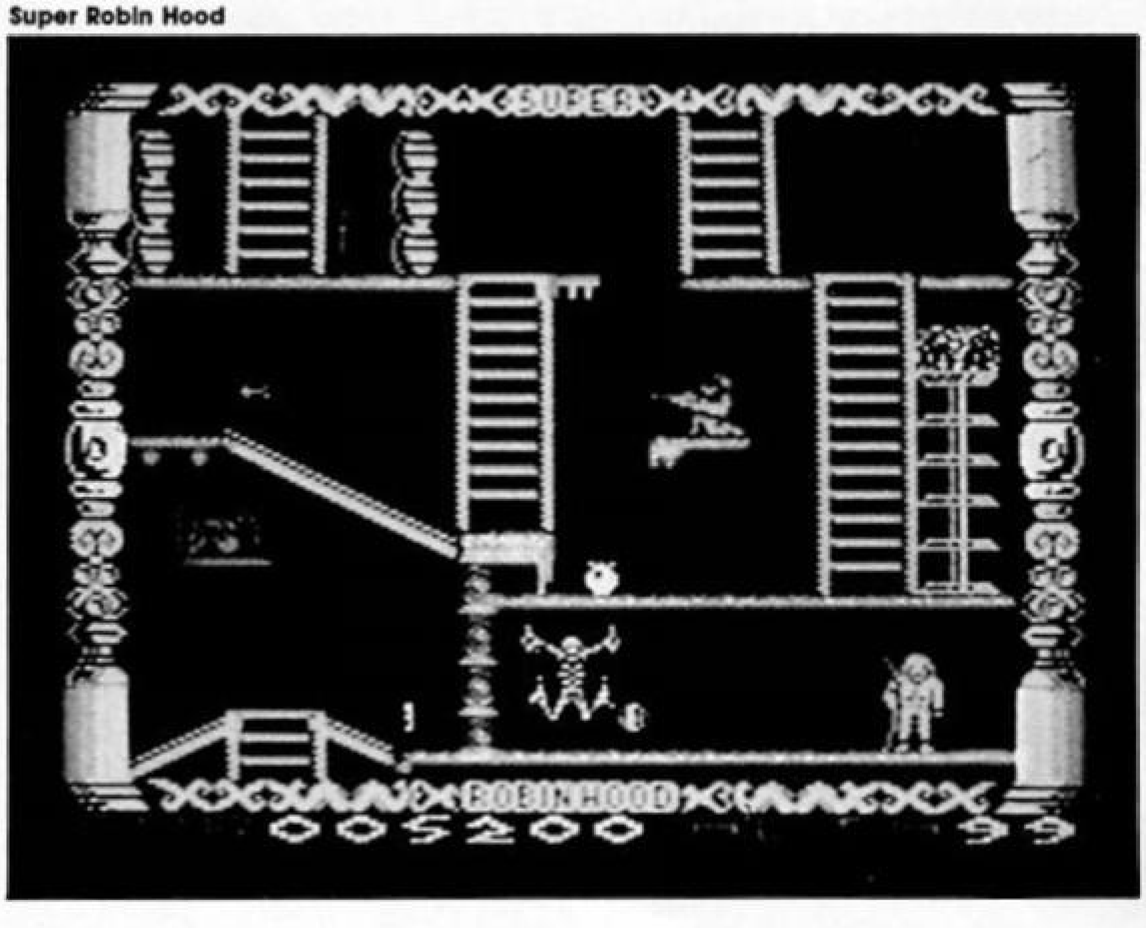
Budget games are accounting for more sales than ever. ZX looks at the latest releases.

■ f the computer press is anything to go by, budget software is taking over the world. Nobody is buying full-priced games apparently (unless they're conversions of coin-op titles), and certainly Mastertronic, if they're not taking over the world, are taking over Melbourne House.

There are a number of software houses competing for the budget market but for the most part it's a three way fight these days. Mastertronic were the first in the field, closely followed by Firebird's Silver range, and these two have been slogging it out enthusiastically for a while now. Recently though, Code Masters, the label founded by a couple of ex-Mastertronic programmers, has made quite an impact with games such as BMX Simulator and Terra Incognita.

This month we've received some new releases from all three of these companies, giving us a good chance to compare a variety of products and take a look at the state of the (budget) art.

Super Robin Hood



Firebird

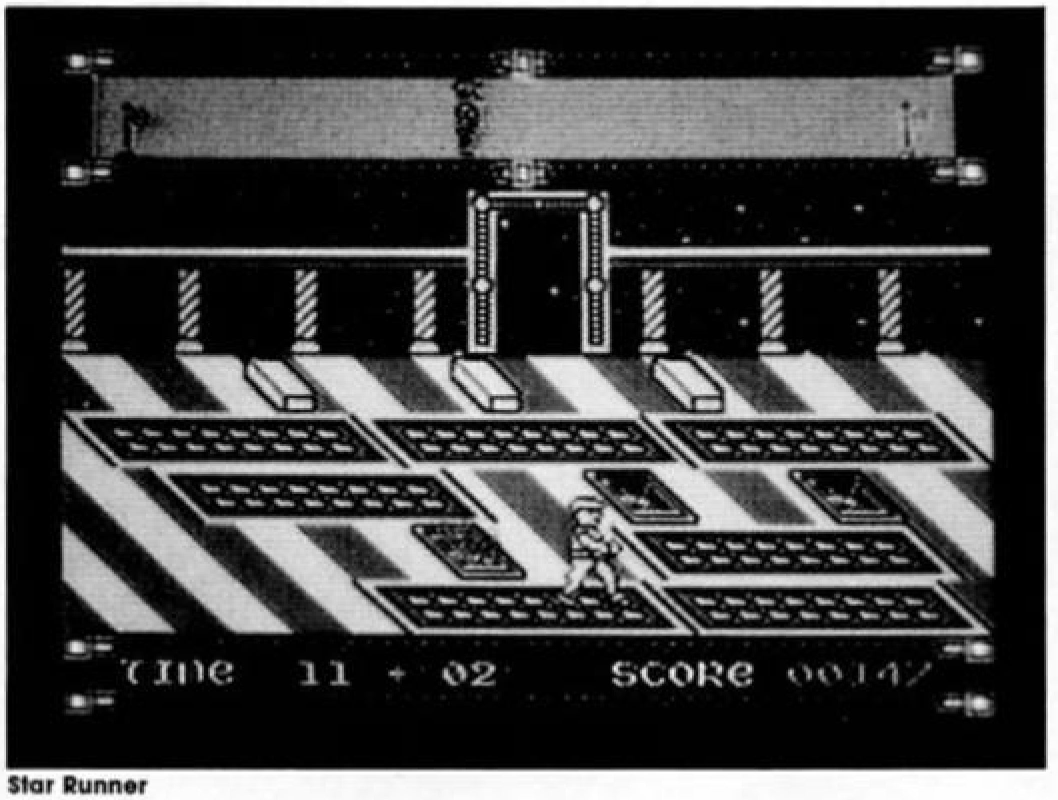
Personally I've not been as impressed with Firebird's stuff as I have with Mastertronic's or Code Masters', but they have been improving a bit lately. Thrust was perhaps their best release for the Spectrum, and Thrust II, whilst basically the same, added a couple of frills to the addictive formula. Their latest game, **Tubaruba,** is a fairly run of the mill effort. You have to find your way around a school building and collect £50 to pay for a broken window. You have to contend with the usual assortment of deadly sprites, ladders and so on, and while this is the sort of thing that could have been sold at full price a couple of years ago it's still a bit bland by today's standards.

Mastertronic

There were quite a few Mastertronic releases just after Christmas, but they've been a little bit quiet lately (probably busy planning to release Melbourne House's back catalogue as budget titles...)

They recently added the £2.99 label, Bulldog, to their stable and got off to a flying start with **Feud** (reviewed last month). **Colony,** the lastest Bulldog game isn't quite so good. It's basically a maze game set on a distant planet in which you have to control an android and protect the crops planted by the human settlers.

Colony is very much in the mould of Mastertronic's £1.99 games, and graphically bears quite a resemblance to many of their other maze games, though for the additional pound the



Star Runner

presentation and control options have been made a little more sophisticated.

Meanwhile, back on the £1.99 label, **Rasterscan** is a bit of an oddity. It claims to be an animated adventure requiring no arcade skills at all. This isn't strictly true as the game puts you in control of yet another android (this time known as MSB), one which looks and moves like a beach ball, so you'll need a little bit of arcade skill to control its movements. Your task is to move MSB around a damaged spacecraft and repair its engine and scanner systems. Along the way you'll be presented with a number of logical puzzles, but as well as solving these you'll actually spend a fair bit of time figuring out just what the point of the game is (at least I had to, anyway). It's not the sort of game that will appeal to arcade addict, but if you're into games that require a bit of thought then Rasterscan could prove quietly addictive.

Code Masters

It's been a good month for Code Masters, with five of their games arriving for review, and, apart from the unfortunate **Brainache** (reviewed elsewhere this issue, along with Transmuter)they're all faring quite well. **Transmuter** is a good version of the old Scramble format, and **Ghost Hunters** ("featuring voice synthesis") is one of the better platform games to come our way recently. Set in a haunted house, the game sets you the task of rescuing someone who has been trapped within the house. There are all sorts of ghoulish sprites out to do away with you, but fortunately you're armed with a ghost busting rifle which evens the odds somewhat. There's nothing at all original about platform games, but this one seems to have been thought out quite well. You use your rifle by moving a hairpin sight around the screen, but while you're doing this your character is immobilised and this adds an extra little edge to the game since you have to be quick when it comes the choice of whether to use the rifle or just run away.

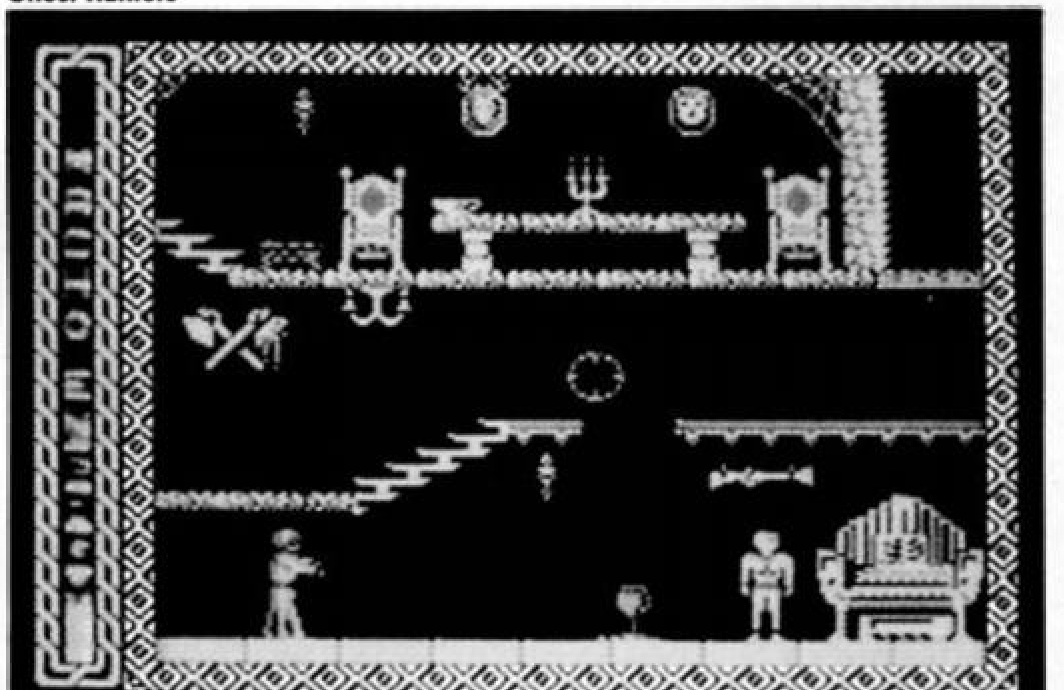
Super Robin Hood is along similar lines, except that it's Robin Hood helping out Maid Marian in a mediaeval setting.

Star Runner is from programmer Christian Urquhart (who's just released a full price game on Hewson's label). It's a simple game involving guiding a running figure along a route dotted with fire pits, stumbling blocks, robot guards and the like. The graphics are quite nice, but the game does seem to be a bit too simple to offer a lasting challenge.

Generally speaking, Code Masters' games seem to be a little less formularised than those of Firebird or Mastertronic who at times seem to be a bit over fond of the dated maze game format. This seems to be paying off for the newer Code Masters and they've done well to establish themselves so quickly whilst other budget labels (even from such big names as Elite and U.S. Gold) have failed to make a dent in the apparent Mastertronic/Firebird monopoly. If I had to pick a couple of games from this month's budget releases I'd go for Transmuters and Ghost Hunters. That puts Code Masters ahead this month, with Mastertronic in second place and Firebird in third.

Still, a month is a long time in budget software so perhaps next month things will change (perhaps with the release of Thrust 3 from Firebird. . .).

Ghost Hunters





MINDPLAY

Adventure, news, reviews and advice, from Pete Sweasey's dungeon.

Normally the supply of adventures dries up at this time of year. Not so in 1987; some big names for review this month, with more on the horizon.

The real biggie is *The Shadows of Mordor*, the inexplicably named sequel to Lord Of The Rings, Game One. This has been long awaited by adventurers, though not necessarily in eager anticipation. More in hope that it would be better than the first game.

The first adventure was, to be frank, pretty awful. It was mind-numbingly slow, not least due to that ridiculous routline of moving pictures of the characters around the screen (might have looked a good idea on paper — but boy was it poor in practice). The graphics were rarer than jokes in ITV sitcoms, and worse quality too (almost Level 9 standard). The text was dull, the 'problems' even duller. The moronic programmers presumably thought it would be funny to add little anachronisms like photographs and references to heavy metal music, but Tolkien fans were rightly incensed. Some interesting features were offered, but the programming had run away with itself, and the game had been swamped by the attempt at technical sophistication (in which respect LOR was not that special anyway, when compared with Level 9's Worm In Paradise from around the same time). You needed remarkable perseverance to reach the end. I did not, incidentally. Add to this an unaffordable (for most people) price of £14.95, part of which paid for a book which many of the potential buyers would already have, and you have a true software disaster. Alan Davis, author of Runestone and fellow ZX contributor, said this about it: "sheer monstrousness . . . oh, it was marvellous for a few days — perhaps a case of the Emperor's new clothes — but now it gathers the dust, unplayed and indeed, unplayable. It's hard to forgive Phillip Mitchell for this . . ." To find out whether he deserves our clemency, read the review.

Before I press on (lots to be done this month) a word about ratings, or rather eight words: do not take too much notice of them! I am a terribly indecisive person — well, quite indecisive — no, very indecisive — anyway, I am rarely certain about which of the grades to give a game. And, of course, games which gain the same rating are not of

identical quality: a 'Great' game could be just a smidgeon away from Monster Hitdom, or almost a 'Good' but saved by a particularly interesting feature. The important point is to read the review; the rating is only a rough guide. The only certainty is that a Monster Hit is of outstanding quality — but even then, it may not appeal to everyone.

Still, enough of these meaningless maunderings. On with the reviews!

EARTHSHOCK
8th Day
£3.99

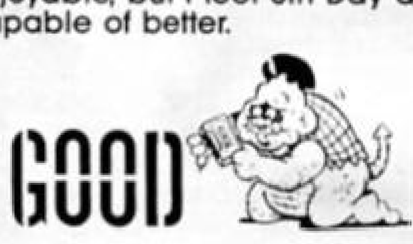
Another title from Eighth Day, pioneers in quality budget adventures, and the people behind the wonderful HRH. This is in their new 'Grafix' range, though pictures are conspicuous by their sparsity, and when they do appear, they are distinctly average.

Earthshock has a lot of text, because it is a comPressed Quilled game. On one side of the tape is the text and graphics version, where the words are pretty verbose, on the other is the text only, where veritable floods of text appear. The writing is generally above average, though sometimes tedious.

The theme is a bit dreary, a futuristic fantasy science fiction thingy, where as a member of a post-apocolyptic tribe you must battle with the droids who have taken over the world. I'm fed up with working out how to use futuristic technology (there always seem to be slots for inserting cards into). Still, there are some nice touches, and the scenario is pleasantly detailed. You receive an admirably full instruction/story booklet, very good for four quid.

The adventure is entertaining enough for budget, though vocabulary is weak (for example, with "a pile of rubble," SEARCH RUBBLE is fruitful, SEARCH PILE and "you find nothing unusual"), and the problems are nothing astounding. An innovative feature is 'three way' play — at one point you make a decision which sends you on one of three sub-quests, and determines one of three end-games. So theoretically, you can play three times. Decide for yourself whether this is good or gimmicky.

Some above-average features, enjoyable, but I feel 8th Day are capable of better.



THE SIDNEY AFFAIR
Infogrames
£8.95

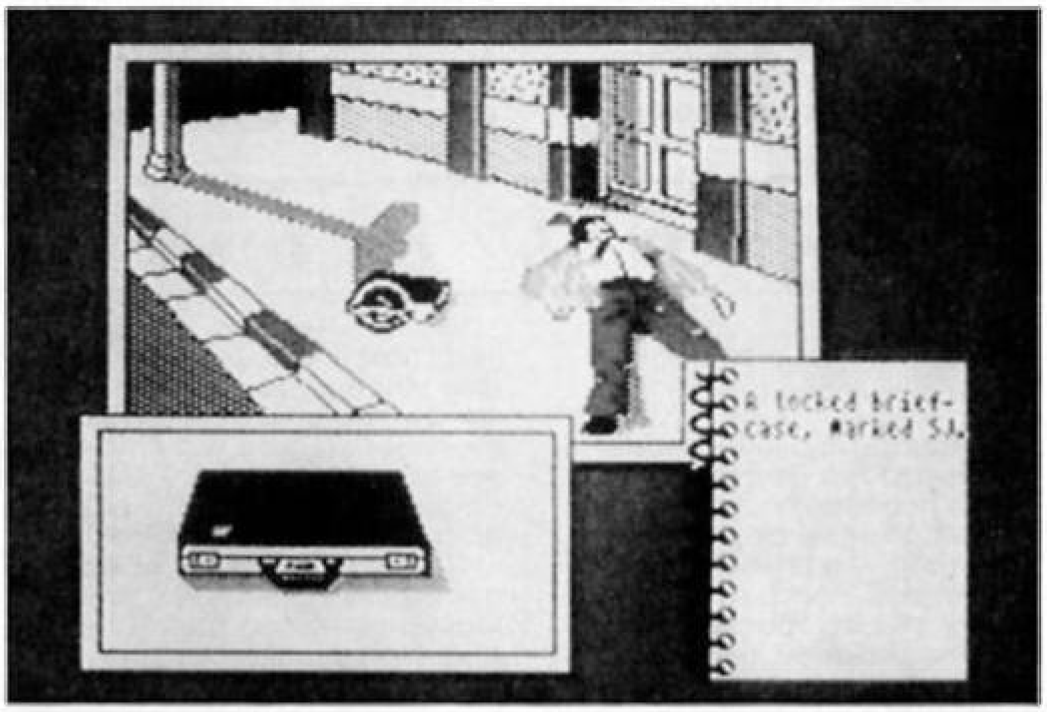
The Sidney Affair, sequel to *The Vera Cruz Affair*, is in a very similar style, and has an unimaginatively similar title. Again you are a French police chief, again there is a grisly murder to be solved, and again you face the even harder task of using the 'Diamond' network, France's police computer system.

As before, the game has two sections. in the first, the scenes of the crime (two of them) are portrayed, with rather neat graphics, and by moving a magnifying glass over them you attempt to zoom in on clues. Here you can find the victim's identity and some other details (check out the briefcase). Part two allows you to take statements, perform an autopsy and ballistic examinations, and (try to) use the infernal computer to gain information on police records and other forces related investigations. Then correlate the evidence and make the arrest. Simple . . .

My thoughts are as for Vera Cruz: really only the case details have changed. The same sense of genuine investigation exists; the joy of detecting leads hidden within statements is worth experiencing. However, when stuck, I'm still uncertain as to whether I'm on the right track but entering the wrong computer commands, or whether what I'm trying to do is irrelevant. I still feel more friendliness is needed, especially as few of us Brits understand the French system of departments and regions, or their formats for things like telephone numbers.

If you like Cruz, you'll like this. More of the same — slick, unusual and a good puzzler.





SCOTT ADAMS' SCOOPS
Master Games
£9.99

Funnily enough, only one of the games on this compilation is written solely by Scott Adams. One is written with a chap called Phillip Case, another is co-written with his wife Alexis, while the fourth is written entirely by her. Still, "Scott-And-Alexis-Adams-with-Phillip-Case Scoops" is a somewhat less engaging title, I guess. The important thing is, all the games here are in that distinctive Scott Adams style we're come to know and hate.

Some of you may never have played a Scott Adams game, in which case consider yourself lucky. Way back in the mists of computing time (i.e. during the late seventies) he was one of the first people to release an adventure for a home computer (a TRS80, or something similar). For this computer's extremely limited capabilities, what he produced was remarkably good, so it sold well. Time moved on, better computers with greater memory arrived, the market changed — but Adams did not. He kept on churning out games which fitted into 8K, converting them for the new computers as they arrived. His name being a venerable one, people bought his adventures, even though he was a dinosaur who should have evolved or become extinct. Critics gave him good reviews out of respect for what he once was, though, if written by anyone else the games would have been ignored. Their only

redeeming factor was the quality of his puzzles — fiendish and ingenious. Unfortunately, the parser was always very limited, the text brief to the point of ridicule, the responses few and sloppy.

He would probably never have succeeded in the Spectrum market, except that his first games released on the Speccy were enticing licensed names — *The Hulk* and *Spiderman* — and that they were accompanied by some stunning graphics, at the time the best the Spectrum had witnessed.

But this US Gold compilation does not consist of those more attractive Adams productions. It contains four TEXT-ONLY adventures, previously unreleased on the spectrum (as far as I know). Three of these were originally written over five years ago; the fourth is based on a film no-one in this country has heard of. Hardly a tantalizing package, is it?

The games are *Pirate Adventure*, a beginner's game with self explanatory theme; *Vodoo Castle*, a supernatural quest to save a Count's life; *Strange Odyssey*, a space adventure which is now unplayably cliched, though it would have been quite good when written; and Buckaroo Banzai. The last really is an oddity. The film is a very weird, apparently incomprehensible science fiction fantasy which completely flopped on its initial American release (around two or three years ago) and has never been seen here, though it has since developed a cult following in the States. Even stranger is the game, which is in the typical curt Adams' style, and is completely meaningless and pointless if you haven't seen the film. Which, of course, you cannot have done. I managed to find out a little about Buckaroo from a cinema magazine I read, but US Gold's utterly worthless instructions tell you just this "Only by unravelling the many puzzles set by Scott Adams do you stand any chance of completing this futuristic adventure." (What John Cleese would call a bleedingly obvious statement. How else do you complete an adventure? By spreading a cheese sandwich over the keyboard?) "Set in the world of pop groups and science fiction." Very informative.

On the Spectrum, Scott Adams without graphics is like a radio version of a Charlie Chaplin silent film. The text is no compensation. Here is an example of Vodoo Castle at its most descriptive: "I am in a room in the castle. Visible Items: A big kettle."

The vocabulary does not even include WEAR or I (though the imbecilically inept instructions claim that it does).

Having read all this, you may be wondering why the game is not getting the lowest possible ZX rating (a 'Groan'). Well, considering there are four games on a £10 tape, they work out as budget titles — £2.50 each. As such they are not entirely bad. The puzzles are fun to crack, and the lack of distracting features (like text, graphics, a decent parser or a noticeable number of locations) tends to concentrate the mind. So for some good problem solving — I realize there are people who like Scott Adams style of games even if don't understand why — this might be worth buying. But anyone else — steer clear. This is adventuring at its worst.



GRIM

HELPLINE

This month's helpline is mostly devoted to two games about which I constantly receive a lot of mail. It seems appropriate to begin with the dreadful *Lord Of The Rings Game 1*. As I said in this month's introduction, I never had the patience, nor the time, to complete this. As a result, a number of helpline requests on it have been piling up in a corner of the dungeon. I'll now attempt to answer them, by using a solution sent to me by Rochdale's master-adventurer, John Wilson. WARNING: these solutions are explicit, step by step guides!

Charles James is trying to complete the quest set by the Monk near Radagast in the first part. When he sets the task, go N, W, N, W, S to find the Green Knight. Attack and defeat him, and take the broken medallion that he leaves behind. Now retrace your steps to the Monk, then go N, E, SE to the red Lady, who you must kill, and whose medallion should also be collected. The monk will now allow you to pass through the stone door, to the south of him, which opens to reveal another door. KNOCK, WAIT and Radagast will appear. He will give you the magic words which release the powers of the pale green jewels, and will enable you to pass the Black Riders (of course, you could just avoid them).

Surviving the mines of Moria is our next task, set by Andrew McMaster. Having said "Friend" and entered, and having attacked the tentacles, go east. Now that you're in the mines, go up, east three times, down twice, west three times, down once and you should be in a corridor. PUSH HORSE and the slab will open. Go east three times, up and east into a room. EXAMINE and READ the things you find here. WAIT for Orcs to appear, bash them. E, WAIT and Gandalf will chat to you. E and LIGHT CANDLE WITH MATCH. DOWN twice, S, D, S to the square room. Make sure your party is with you; if not, backtrack to collect them. Now S, D and proceed E until you reach the bridge. Something thrilling happens! When told to flee, continue east and you'll emerge in a dale. This is only a basic route through; there is more to do if you want to.

Help in the Lothlorien area, for A. Jobber, and Nicolas Claux from Portugal (ZX is read worldwide). Having arrived at the eastern bank on the Nimrodel, an elf will enter. WAIT, then proceed EAST to the river. Here, the best policy is to SAY TO ELF "HELP", and to SAY TO GIMLI "GO NW" (since they don't get on). When he has gone, you must go NE, over the rope, and WAIT. W, WAIT again. When the doors have opened, head E to the tree, then UP, and SAY TO GALADRIEL "HELP."

Questprobe

On to Questprobe 3. Very few people have solved it, not least because many of the copies seem to be bugged. I am indebted to Malcom Harden who has sent me a complete solution. I have not been able to check it fully — my copy is a pre-production one, so probably in even less working order than the rest — but I'm sure it's all correct. Marcus Beer wants to move Blob. This should be done towards the end of the game. You need the cannon from the circus tent, and the gunpowder from Latveria. Outside the castle, as Thing (must have taken someone ages to create that name) LOAD CANNON, GO CANNON. Switch to Human Torch and AIM CANNON AT BLOB. FLAME ON HIGH, LIGHT FUSE (it begins to burn). THROW FLAME AT

BLOB — he will dodge to avoid it while Thing will be fired into the wall he was blocking.

Nicolas Claux has managed to gain access to the tunnels for Thing, but is now stuck. From the wall of fire where you light the candle, go W, S, S, E, S. You'll find a place where air is sucked in — represented in the astonishingly verbose Adams style by the word "Woooooo." SWITCH to the Torch, fly up to the cave (FLAME ON HIGH). REST for a bit, then THROW HIGH FLAME AT BOULDER. This chips off a pebble. EXAMINE BOULDER, and, together with the re-appearance of the "Woooooo" noise, it would seem Torch is at the top of the shaft we left Thing under. PUSH PEBBLE UNDER BOULDER, SWITCH and WAIT until it appears at Thing's end. Then he can THROW PEBBLE HARD UP. As Malcolm says, NOT the most obvious of sentence constructions . . . This should also help Andrew Bennett.

Marcus Beer is also freezing to death in the coach in Dracula (cheap tour company, obviously). LOOK AROUND and you'll find a seat. LIFT SEAT, EXAMINE DOOR — locked! So, REMOVE CROSS, INSERT CROSS and TURN CROSS, an improvised key, you see! OPEN DOOR, LOOK and you'll find some blankets. Gavin Woods needs to scale the dark brick wall in the third part of the same spinechiller. If you head E then S from the bricked up gate, and LOOK AROUND, you will discover a tree. Climb it to see what's on the other side — too big a jump, however. So CUT TREE with the axe you found in the small hut (by smashing the window). Now if you climb tree, you will be on the top of the wall. Head west twice, then north, and LOOK AROUND for a means to reach the ground. A little hint though — once you cut down the tree, find someone who wants the axe. They will be most helpful in return.

Finally, brief aid for Gordon Vooght with The Vera Cruz Affair. To examine writing, enter GRAPHOLOGICAL as an examination. You cannot find any information on Georges Lerat.

Summertime Blues

The days are reaching their longest, the sun shines bright, the beaches beckon, and the sound of sneezing hay fever sufferers fills the air. Yes, summertime is getting into full swing. And are you out their enjoying it? "No!" you cry. Why not? "Because," you scream in anguish, "we're hunched over our computers and won't leave until we solve this blooming adventure game. We're almost at the end — but we just can't get the precise wording."

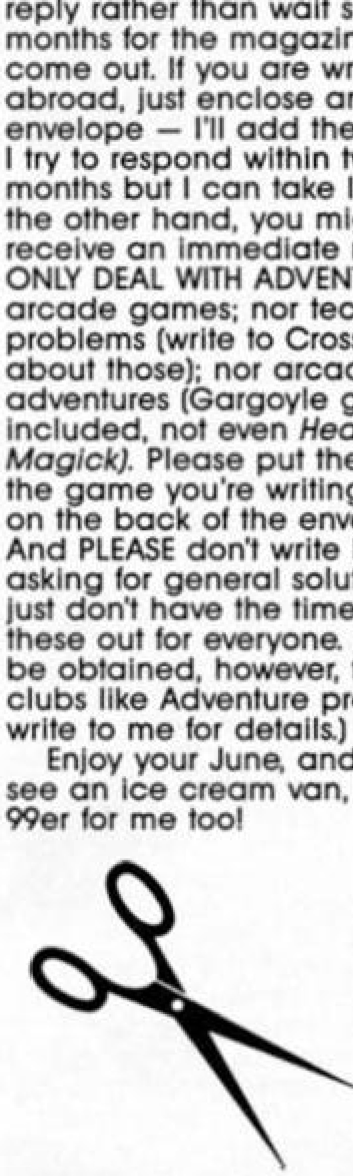
Now, now. Leave the flickering screen and go get some sunstroke, sand in your toes and skin cancer. Let me do the worring, let me be the pale one, let me solve your adventuring problems!

Fill in the coupon printed here, read the small print below, then post it to Mindplay, No 1 Golden Square, london W1R 3AB, and before you can say "Phew! What a scorcher!" or "McGaskill was wrong AGAIN!" you will know the answer.

British correspondants, please enclose a stamped, addressed envelope if you want a personal reply rather than wait some months for the magazine to come out. If you are writing from abroad, just enclose an envelope — I'll add the postage. I try to respond within two months but I can take longer (on the other hand, you might receive an immediate replay). I ONLY DEAL WITH ADVENTURES. Not arcade games; nor technical problems (write to Crosswires about those); nor arcade adventures (Gargoyle games included, not even *Heavy on The Magick).* Please put the name of the game you're writing about on the back of the envelope. And PLEASE don't write letters asking for general solutions — I just don't have the time to write these out for everyone. (They can be obtained, however, from clubs like Adventure probe — write to me for details.)

Enjoy your June, and if you see an ice cream van, buy a 99er for me too!





Title: ....

Company: ....

Problem: ....

....

....

I can help solve: ....

Name: ....

Address: ....

....

Shadows Of Mordor

Another big adventure release this month, Lord of The Rings, part II.



THE SHADOWS OF MORDOR
Melbourne House
£7.95

The good news: *Shadows of Mordor* is a vast improvement on *Lord of The Rings Game 1*. The bad news: it is still not as good as it should be.

The game follows the story of the second book in the Rings Trilogy, *The Two Towers* (so why wasn't it called that?), and is the continued quest of the hobbit Frodo, and his friend Sam, to dispose of a magic ring before the evil Sauron gets his hands on it, or rather his evil finger through it.

Screen presentation looks familiar — the same style as game one except those "margins" which housed the irritating faces have disappeared, as, thankfully, have the visages themselves. In fact the game is text only, though as a peculiar freebie, the B-side contains what amounts to a middle earth Travel Brochure; many scenes from the game are illustrated in a flick-through gallery. Though simple, some of these views are highly attractive, and it's a shame they were not included in the main game (apparently they will be on the 128 version).

The descriptive text is sadly lifeless and mechanical, a typical location being "Frodo is on a cold, windswept ridge in a range of dry hills." Not really an advance on The Hobbit, is it? The text is made to look long by listing, in a clinical manner, who is there, what those people are carrying, and any objects (these are always referred to with a couple of adjectives, as in "the heavy round rock.") The text has no variety, no passion — and so lacks atmosphere, surely a vital component of Tolkien's work.

Then we have the parser, the performance of which Melbourne House expects should astound us. It's true that you can do some clever things, however when the parser does not understand one of the words in the input, it refuses to do any of it. This means if you type in a string of commands, the last of which is not understood, none of them are performed. It also means you cannot simulate natural conversation, for example saying SAM "GIVE THE BOOK TO ME PLEASE", because words like PLEASE are not understood. In most games, the action would be performed anyway. The parser here is too clever for its own good.

The vocabulary is large, but with some curious omissions. Character interaction is bound to be limited if the game does not understand any question words. And you are occassionally expected to enter too obscure combinations, such as POLE RAFT (a clue!).

The most endearing feature of *Shadows of Mordor* is its problems, which involve complex object manipulation. I especially liked the lack of fixed solutions — there are often alternatives. For example, to descend the cliff at the start, the best way is to use some rocks, but I devised a way whereby Sam would uproot a tree, and I would go down the hole this created. This method works, though unfortunately it also causes Sam to be splatted! Few other games would allow you to vary and experiment in that way — very commendable.

So, *Shadows of Mordor* is cheap, complicated and fun to play. However, I strongly feel it does not do Tolkien's work justice. I can't help thinking that someone like Level 9 could produce an equally sophisticated and far more friendly program, with more to do, and much more atmosphere and subtlety. Only a program like that would truly capture the magic of The Lord Of The Rings.

Pete Sweasey





INDOOR SPORTS

A quartet of indoor games from Advance.

Indoor Sports
Advance
£8.95

Four games for the price of one is the claim to fame of this unusual compilation of indoor sports. You can hang up your skis and running shoes for this one as they are indoor games that everyone must have played at some time in the past.

The gametape contains a main program that includes a menu from which you can load in the four events. Although they can be played in any order it's best to tackle in the order they appear on the tape.

Air hockey is without doubt my favourite game as I've spent many an hour in the arcades smashing my knuckles in an attempt to drive the puck into my opponent's goal. In the real game the puck is supported by a cushion of air and can reach some incredible speeds when it can zip off the table and hit some unsuspecting space invader. The computer version of this doesn't have that important feature but it does offer four computer opponents, three game speeds and the smoothest graphics you are ever likely to see.

Darts has never been done well on a computer and this one's as good, if not better than most. The event consists of two screens, one to aim the dart and keep score and the other is a graphic view of the player throwing the dart. The first screen is the business end of the game and is where the players decide the direction, angle and strength of each dart. After the throw which is shown on the other screen the computer chalks off your score and the game continues until either one player gets the double to finish a 301, 501 or 701 game or a player runs out of time that ticks away at the bottom of the screen. This version of darts is timed to prevent slow play!

Table tennis has never been a great skill of mine but now you can play, get hammered but without the embarrassment.

You can play either a single match or a series up to the best of seven matches. The games can be played at beginner, normal (fast) or hyperdrive (very fast) speeds so it's wise to opt for the additional auto move option that automatically moves your bat to meet the ball leaving you to mis-time the shot. Take on too much straight away and you'll certainly get beaten by a ruthless opponent.

An additional feature allows you to bias a match by assigning between 6 and 20 skill points to each player and can be distributed between smashing, forehand, backhand, reactions, speed and endurance factors that effects the movement and skill of the ghost like bat that hovers above the table.

As with all the events the attention to detail is extra-ordinary, particularly the subtle difference in shading of the ball's shadow, that makes it look like a reflection in the polished table top.

The fourth and final event will leave you skittled as the ten pin bowling is probably the hardest game to learn and master.

The game begins with a side view of our bowler competing against human or four level computer opponent as he begins his run up. By selecting joystick positions you can stand him anywhere in the lane and select the general direction you want the ball to go in. The tricky bit is timing a final joystick press as he swings the ball to release it, as if you press it too soon it will drop on his foot for a foul. Too late, and he'll follow it down the alley as he takes a nosedive!

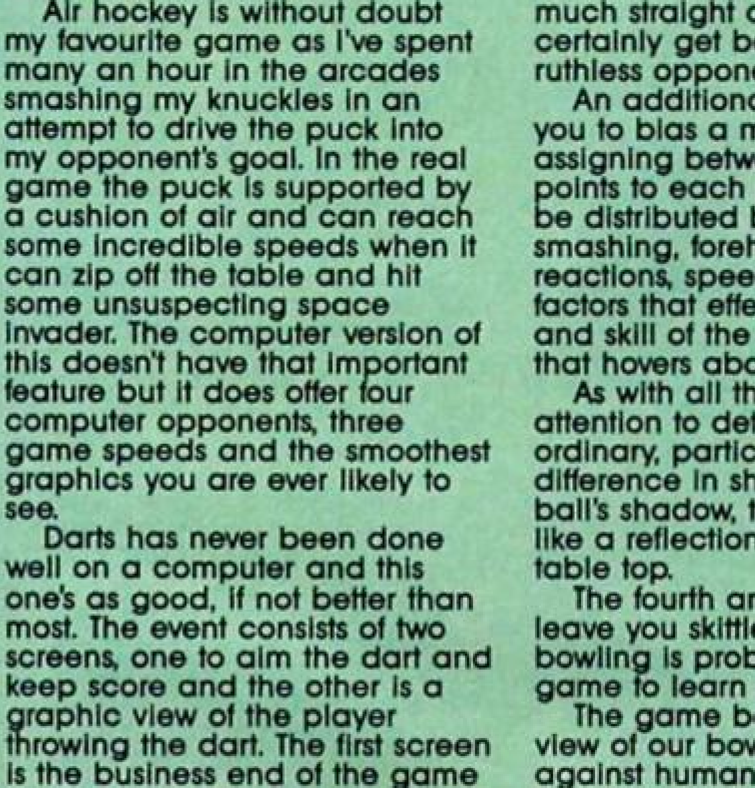
To add to your problems you can also curve the ball into the skittles if you know what you're doing but it's more likely it'll end up in the gutter.

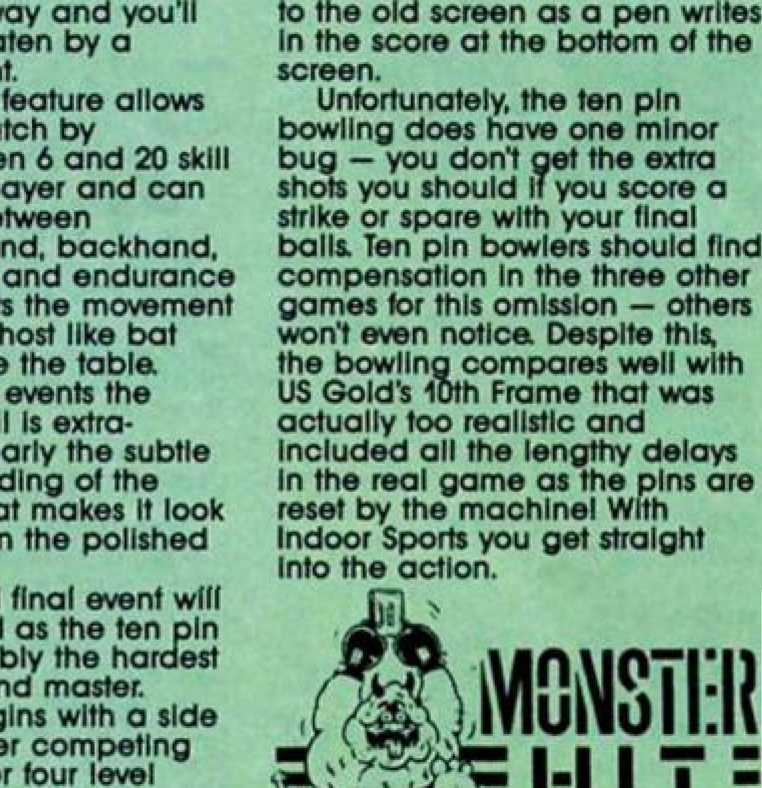
The scene now shifts to a head-on view of the skittles as the ball careers down towards them. Above the lane is an added comic touch as the game mimics your facial expression as the ball either skittles the lot for a strike or falls uselessly into the gutter.

The game then switches back to the old screen as a pen writes in the score at the bottom of the screen.

Unfortunately, the ten pin bowling does have one minor bug — you don't get the extra shots you should if you score a strike or spare with your final balls. Ten pin bowlers should find compensation in the three other games for this omission — others won't even notice. Despite this, the bowling compares well with US Gold's 10th Frame that was actually too realistic and included all the lengthy delays in the real game as the pins are reset by the machine! With Indoor Sports you get straight into the action.

MONSTER HIT





Comic Strip



RANDOM MEMORY

Clyde Bish finishes his comic strip utility.

So far in this series we've looked at ways of quickly producing a cartoon-style picture, with speech bubble and caption on screen, using the minimum of memory. Now it's time to get things moving, and the first thing we have to move is the existing strip one picture to the left to make screen space on the right for the new illustration. The code from Table A will do this for you. Load it in, reading across each line, using Program 1, and SAVE with:

SAVE "LSCROLL" CODE 63905,114

and VERIFY.

If you want to try it without loading in a picture type:

FOR f = 1 TO 250: PRINT f: NEXT f (ENTER)

to get a block of numbers at the top of the screen, **RANDOMIZE USR 63905.** You'll see the eleven rows scroll eleven columns to the left, and a bar appear at column 21. To the right of the bar is a blank space. This is where the next picture and caption will be drawn.

Before we go any further, try this out:

PRINT "O"; (ENTER)

What happens? Apparently nothing. Actually you have printed a 'O', but it's in the same colour INK as the PAPER, and so invisible, because the LSCROLL routine ends by setting INK to

Program 1.

```
 10 CLEAR 63617: FOR f=63905 TO
 64010: INPUT i: PRINT f,i: POKE
 f,i: NEXT f
```

PAPER colour. You can prove that it's there with:

INK 0: PRINT "O" (ENTER)

and you'll see a '0' to the right of the bar with a space before it. That's where the first 0 is!

Captions

You may be wondering what the point of this is. The reason is that it enables us to draw the next picture and caption invisibly, then suddenly make it appear. How? Use program 1 (with the FOR . . . TO values altered to 63858 and 63904) to LOAD in the code from Table B. This new code will alter the attributes (the data file in the machine's memory which controls screen colours) in the picture box to black on cyan, and those in the caption box to black on yellow, as well as resetting global INK to black for subsequent PRINTs. In the final control routine you'll be able to choose the picture box colours but for the moment leave well alone and try both code blocks out together with Program 2. You'll see the screen fill with numbers, then scroll. The PRINT lines will suddenly appear in the boxes, followed a moment later by the INK reset message. This will repeat. Press BREAK when you get bored.

Program 2.

```
  5 REM LSCROLL & ATTR CODES
     must be on board
  10 BORDER 0: FOR f=1 TO 250: P
RINT f;: NEXT f
  15 PRINT AT 11,0;"

            "
  20 PRINT #0;"BREAK TO END": PA
USE 50
  30 RANDOMIZE USR 63905
  40 PRINT AT 2,22;"**********";A
T 3,23;"NEW PIC";AT 4,22;"*******
***";AT 9,23;"CAPTION";AT 10,23;
"*******"
  50 RANDOMIZE USR 63858
  60 PRINT AT 12,8; BRIGHT 1;" I
NK RESET TO 0 "
  70 PAUSE 80: PRINT AT 12,8;"
           ": GO TO 30
```

Before you go any further SAVE a copy of the attributes code with: **SAVE "ATTR"CODE 63858,47** just in case you get your electricity cut off!

Now we have our pictures and captions on screen and moving, it's about time we thought about how we are going to hold the data addresses etc. We will need to POKE into the various machine code blocks that will be doing the work for us. Let's start by making a list of what we need to know. We need the start address of the draw CODE, preferably held in low/high byte form. We need a similar address for the string/icon/bubble pointer data start. (In future we'll just call this the rest CODE). We'll need the value of the caption text pointer — the start position of the caption in m$, and finally the value of the attribute to fill the picture box. The cheapest way

Program 3.

```
   5 LET s=2: PRINT #0;AT 0,0;"H
ard Copy?": PAUSE 0: IF INKEY$="
y" THEN  LET s=3
  10 INPUT "No. of pictures?",n:
 DIM p$(n,7,3)
 100 FOR f=1 TO n
 110 INPUT "DRAW data start: pic
ture ";(f);"?"'a: RANDOMIZE a: L
ET p$(f,1)=STR$ PEEK 23670: LET
p$(f,2)=STR$ PEK 23671
 112 PRINT #s'"Picture ";f;":"
 115 PRINT #s;"DRAW start",PEEK
23670;" (";a;")"',PEEK 23671
 120 INPUT "REST data start: pic
ture ";(f);"?",a: RANDOMIZE a: L
ET p$(f,3)=STR$ PEEK 23670: LET
p$(f,4)=STR$ PEEK 23671
 125 PRINT #s;"REST start",PEEK
23670;" (";a;")"',PEEK 23671
 140 INPUT "Text pointer: pictur
e ";(f)"?"'a: RANDOMIZE a: LET p
$(f,5)=STR$ PEEK 23670: LET p$(f
,6)=STR$ PEEK 23671
 142 PRINT #s;"POINTER:",PEEK 23
670;" (";a;")"',PEEK 23671
 145 INPUT "Attribute No.: pictu
re ";(f);"?"'a: LET p$(f,7)=STR$
 a
 150 PRINT #s;"ATTRIBUTE:",a
 200 NEXT f
 210 SAVE "picdata" DATA p$()
```

to hold this information, byte-wise, is to use a string array set to three dimensions: the number of pictures to be called, the number of bits of information to be held (high/low for drawCODE, restCODE and pointer, and attribute value, i.e. 7) the maximum length of any bit of information (a three figure number).

So, our array will be dimensioned p£(n,7,3) where n = number of pictures.

Editor

Having decided on the format, we now need to get the information. Program 3 is an editor designed to do the job for you. When you're ready type it in and RUN.

You'll be asked first of all whether you want hard copy (y/n), then for the number of pictures you need to cater for. This will set the array and start up the loop. As the loop runs you'll be asked for the drawCODE start address of the picture. Enter the five-figure decimal address and the program will convert it to its low/high byte form and store it in p$(n,1) and p$(n,2).

Follow this with the restCODE start address (to be stored in p$(n,3) and p$(n,4),THE CAPTION START POINTER VALUE (stored in

Program 4.

```
9000 POKE 23296,VAL p$(n,1): POK
E 23297,VAL p$(n,2): POKE 64942,
VAL p$(n,3): POKE 64943,VAL p$(n
,4): POKE 63859,VAL p$(n,7): LET
 p=VAL p$(n,5)+256*VAL p$(n,6)
9001 RANDOMIZE USR 63905: RANDOM
IZE USR 65197: RANDOMIZE USR 649
36: FOR f=1 TO VAL m$(p): LET z$
=CHR$ (f+7)+CHR$ 29+m$(p+1 TO p+
12): PRINT : RANDOMIZE USR 64787
: LET p=p+12: NEXT f: RANDOMIZE
USR 63858: RETURN
```

p$(n,5) and p$(n,6) and lastly the attribute value (stored in p$(n,7). To calculate this enter PAPER * 8 + INK. The loop will repeat for each picture up to the value set for n, before SAVEing the DATA array under the title "picdata".

It may have crossed your mind that there may be occassions when you don't want any background drawn — perhaps when a character is

Program 5.

```
8999 REM USCROLLCODE must be on
         board
9000 GO SUB 9999: LET a$="": PRI
NT AT 20,1: FLASH 1;"+"
9010 FOR d=1 TO 20: NEXT d
9020 IF INKEY$="" THEN  GO TO 90
20
9030 IF CODE INKEY$=12 THEN  IF
LEN a$>0 THEN  PRINT AT 20,LEN a
$+1;CHR$ 8;;" ";CHR$ 8;: LET a$=
a$( TO (LEN a$-1)): GO TO 9010
9040 IF CODE INKEY$=12 THEN  GO
TO 9010
9050 IF CODE INKEY$=13 THEN  GO
TO 9090
9060 LET i$=INKEY$: IF i$>="0" A
ND i$<="Z" THEN  LET i$=CHR$ PEE
K 23556
9070 LET a$=a$+i$: PRINT AT 20,1
;a$: IF LEN a$=30 THEN  GO TO 90
90
9080 GO TO 9010
9090 GO SUB 9999: RANDOMIZE USR
63810: RETURN
9999 RETURN : REM this line is
replaced
```

conversing in close-up. What we will do here is to craftily PLOT and DRAW a point within the black picture frame so that it won't show. What is even more crafty is to use 16235 as the drawCODE start address. "Hold on," I hear you say. "That's in the ROM!" Yes, I know it is, but that doesn't matter. It just so happens that at this address there is a series of bytes 84, 84, 84, 84, 0 which is just what we need! Similarly, if you only want a line drawing set restCODE start address to 64019 — There's a string of 0s there. As for no caption, remember I suggested you made the opening substring of m$ twelve blank spaces? It is that pointer value you give.

In order to run all the codes in the correct order you'll need to include Program 4 as a subroutine within your adventure program. To change pictures you set into the picture number then call the routine. Line 9000 passes the values stored in p$() to the various machine code blocks whilst the present picture in still on screen. Line 9001 begins the change, scrolling the picture, calling the DRAW and REST routines, and using the values in m$ to produce the caption. ATTR make all appear from nowhere before the routine RETURNs.

So now we have all the codes and controls needed to produce and move our pictures and captions, plus scroll speech bubbles, but that's not all we need to be able to do. Next we

Program 6(a)

```
   1 DEF FN m(x,y)=(x+y+ABS (x-y
))/2: DEF FN u()=(65536*PEEK 236
74+256*PEEK 23673+PEEK 23672)/50
: DEF FN t()=FN m(FN u(),FN u())
: REM include this in set up
  10 POKE 23674,0: POKE 23673,0:
 POKE 23672,0: REM this sets
timer at start
```

Program 6(b)

```
9999 LET t=FN t(): LET h=INT (t/
360): LET t=t-h*3600: LET m=INT
(t/60): PRINT AT 12,2;"Time:";AT
 13,2;0 AND h<10;h;":";0 AND m<1
0;m;" ": RETURN
```

need an INPUT to the main screen routine in response to the program flow, and ideally one which will scroll those inputs up within a communications window so we can keep some record of passing events.

First alter Program 1 so the FOR . . . to values are 63810 and 63857, then RUN and enter the values from Table C. This is the code for scrolling the communications window up one character square. SAVE with SAVE "USCROLL" CODE 63810, 48.

Now type in Program 5 (which we will add to later) and RUN. You'll see a flashing cursor and line 20. Type something in. Try deleting. When you've said enough (30 characters is the limit) press enter. Your input will scroll up one row. If you repeat the input test five more times you'll see the top message more up each time until, on the last move, it disappears, having scrolled out of the box.

Clock

Well, those are the main routines you'll need, but there are many

Program 7(a)

```
   1 REM set s to start length &
  PRINT AT "       "( TO LEN
s) to  set up
   2 REM call 9000 to increase
              9010 to decrease
9000 IF s=10 THEN  RETURN : REM
 max length
9001 LET s=s+1: PRINT AT row,col
+s;" ": RETURN : REM increases
9010 IF s<0 THEN  RETURN : REM
 minimum length
9011 PRINT AT row,col+s;" ": LET
 s=s-1: RETURN : REM decreases
```

Program 7(b)

```
   1 REM set s to start length &
 PRINT AT row,col+s;"        "
( TO LEN s/8) to set up
   2 REM call 9000 to increase
              9010 fo decrease
8999 REM f loop determines width
of bar. For single pixel omit
control loops.
9000 FOR f=1 TO 3: IF s=100 THEN
  RETURN : REM max length
9001 LET s=s+1: PLOT s,88: DRAW
0,7: NEXT f: RETURN :
 REM increases
9010 FOR f=1 TO 3: IF s<0 THEN
RETURN : REM min length
9011 OVER 1: PLOT s,88: DRAW 0,7
: OVER 0: LET s=s-1: NEXT f: RET
URN : REM decreases
```

others you could use. For example, how about incorporating an on-screen timer into the input routine? This is quite easy to do using the computer's on-board clock. Its values are held in a stystem variable called FRAMES, which counts the passage of time in 1/50th seconds. Here's how to make it work. Hopefully you've still got Program 5 on board as part of this routine joins with it. Add the lines from Program 6(a), which sets up functions and resets the FRAMES system variable to 0, and the new subroutine Program 6(b). Now each time you GOSUB to, or leave, the input routine you will update the timer. Of course you can update the timer at any time just by calling the subroutine. You could, for example, also update the timer by including a call within the picture subroutine listed earlier. The principle of the timer is quite simple. If you want an explanation of the functions look up p.99 in your manual. The remainder of the routine takes that value, t, and converts it to hours and minutes before printing to screen in the Time box. As the timer is set to 0 at the start you can check the value of it to see if a particular event has taken place within a given time limit.

Usually within an adventure you have a limited amount of strength or whatever which is decreased by conflicts or time and increased by food and spells etc. It would be useful to know how you were faring so let's arrange for an on-screen bar chart of strength levels. There are two ways to do this. Which you choose depends on whether you want to use character-sized blocks or pixel width bars and so get a finer graph. First the block type. Program 7(a) is called at line 1000 for a decrease in strength (and subsequent decrease in bar length) and at line 1010 for increase in strength. There's no requirement to use graphic blocks as indicated in the program. You might prefer a half bar, or in fact any character (e.g. a £ sign for money) or design and use a udg. If you POKE the date 102, 153, 129, 129, 66, 66, 36 & 24 into a udg memory block you will have a heart shape to use for popularity or love.

The pixel-bar type is shown in Program 7(b). It uses PLOT and OVER 1 to produce the graph. If you call the routine more than once (by using a loop within it) you can produce wider bars.

How, how about a schizophrenic hero who, like Superman et al, can change persona? It is important for the player to know who he is at any time and this could be shown as a graphic in a box centre right

Program 8.

```
1000 POKE 23675,130: POKE 23676,
240: GO TO 1010
1001 POKE 23675,226: POKE 23676,
240
1010 PRINT AT 12,26;"ABCD";AT 13
,26;"EFGH";AT 14,26;"IJKL": RETU
RN : REM enter caps in G mode
```

of screen. First we need the graphics data in memory. As the block is 4 x 3 characters, this means 96 bytes of data for each of two graphics. My schizophrenic oscillates between a handsome prince who turns into a frog. (Well, what do you think this is? A fairy story!) The data for these graphics is held above the USCROLL code from addresses 63618 to 63809. If you want to see how this routine would work put in the data from Table D between these addresses. (If you also LOAD in all the other codes and data you have saved — ALLCODE, LSCROLL, ATTR & USCROLL) you can save them now as a single block with

SAVE "STRIPCODE" CODE 63618, 1758

but don't forget to CLEAR 63617 before loading it!)

Now to see the Prince transform. We're going to use PRINT AT with udgs "a" to "1". "Just a moment!" I hear you cry. "UDGs start at address 65368, not where we've loaded in the data!" Quite right.

So, before we can PRINT anything we have to convince the machine that the udgs begin where we've put the data. We do this by PODEing the System Variable UDG with the low/high bytes of the address we are using. Program 8 will do this before PRINTing the appropriate persona. You need to call it at line 1000 for the Frog and line 1001 to change this amphibian into the handsome Prince.

To get the main program to call the correct routine is easier than you think. All you need to do is to LET z = 0 when you initialize the program, then include the subroutine

LET z = z = 0: GOSUB 1000 + z: RETURN

to call when you want to make the change. That line may not make much sense to you but it does to the computer. It uses what is called logic. If z = 0 then the statement following the first = sign is true, so z will become 1 and the PRINT routine at 1001 is called. The next time, (z = 0) is false so z is reset to 0. Understand that or not I assure you it works. Try it out! Obviously the value of z indicates the current persona which may influence your characters capabilities and actions. (Beautiful Princesses only kiss frogs, not handsome Princes!)

Program 9.

```
  10 FOR f=0 TO 10: PRINT AT f,0
;"
        "
  20 NEXT f
  21 PRINT AT f,0;"

           "
  25 FOR f=8 TO 10: PRINT AT f,0
; PAPER 6: OVER 1;"
                    ": NEXT f
 200 PLOT 7,40: DRAW 240,0: DRAW
 0,-41: DRAW -240,0: DRAW 0,41
 210 PRINT AT 12,26; BRIGHT 1;"
   ";AT 13,26;"    ";AT 14,26;"
   "
 220 PRINT AT 12,1;"Time:";AT 13
,1; BRIGHT 1;"00:00";AT 12,7; BR
IGHT 0;"Strength:";AT 13,7;"Stam
ina:";AT 14,7;"Luck:"
 230 PLOT 208,79: DRAW 0,-24: DR
AW 33,0: DRAW 0,24
```

Now we have all the general routine on board let's turn our attention to some specific examples you're sure to need.

LOOK
This is where you've forgotten where you are and want to see the picture and caption again. Try something like a close up of your hero with a "WHERE AM I?" bubble, but no caption, followed by a second close-up picture with the caption "You look carefully around." Then go back to the original location picture and caption.

WAIT
Here you could use a close-up of your hero with the bubble "I'M WAITING." and the caption "Time passes . . ."

EAT
As well as a close-up and a bubble such as "THAT'S YUMMY." don't forget to increase the strength variable and on-screen bar, plus perhaps a caption "Your strength increases."

CAN'T DO THAT
A picture for general use in this sort of situation would be a hero close-up with the bubble "SORRY, I CAN'T DO THAT."

INVENTORY
Try this sequence :
a) Close-up and caption "You look on your person."
b) Kneeling icon with rucksack and the caption "You look in your rucksack."
c) If there is nothing to list use a close-up with a "NOTHING" bubble, otherwise print up (in a sequence of frames if necessary) what you have. (You could use the PRINT machine code for this if you reuse the routine within line 9001, altering the PRINT positions.)

There are many other actions and situations you'll need to control. Those I'll leave up to you. After all, it is YOUR adventure!

Now to end, a hints and tips section for those knowledgeable and/or brave readers who want

to alter the screen layout.

Program 9 will give you the basic screen layout my routies work on. (Fig. 1 shows the layout reproduced on a character square grid). Changing some of this is easy. For example, the position of the clock, strength bars etc. and persona icon (if you want them at all) can be changed by altering PRINT AT positions once you search them out in the subroutines. Less simple are alterations to machine code routines, so I'll need to check these out in more detail.

Picture Size

This is set at ten character squares wide. To change this you need to POKE 63861 and 63872 with the new width, and 63906 with the new value + 1. The vertical bar printed by LSCROLL is of graphic blocks. To change the character (you could design a udg), POKE the character code into 64008. It also defaults to black INK. If you want a different colour (to match, say, a blue border) include INK i at the start of Program 4, where i = INK colour. You can also increase the caption area by POKEing 63873 with the height. (You may also need to make alterations in the number of lines allowed in the text editor.) Don't forget to lengthen the bar printed by LSCROLL. POKE 63998 with the new value. If you're not using black ink on yellow paper you'll also have to POKE your attribute value into 64014 and 63881.

As written the routine produces a two-tone picture — INK and PAPER. If you want to introduce more colour or other attributes you can do this using a line such as

PRINT AT row, column; *OVER 1; INK/PAPER/BRIGHT/FLASH* number; "spaces equal to the number of characters you wish to affect"

after the Program 4 routine has RETURNed. So, in the first screen dump shown with the troll, you could make the mystic symbols over the open door flash bright blue ink on yellow paper with

PRINT AT 1, 15; OVER 1; INK 1; BRIGHT 1; PAPER 6; FLASH 1; "(2 spaces)"

You would save bytes by having your opening screen layout load in as a SCREEN$. If you did this you could have a whole screen-width picture to begin, changing to the standard ten characters wide with the first scroll.

Communications Window

This is set to 30 columns wide by five rows high, with top left corner at row 16, column 1. To alter the number of rows POKE 63812 with the new value — 1, and 63814/5 with the new D FILE start address. Use Table E to find this.

Obviously you *can* make other changes but these will involve drastic changes to the code. On your own head be it!

Well, that's STRIP complete. Now engage brain and set to work.

Eat your heart out, Dan Dare!

Table A.

14	11	197	17	31	11	33	0
64	213	193	197	229	84	93	126
35	8	175	71	185	40	2	237
176	62	0	18	225	36	124	230
7	32	231	17	32	248	25	125
254	32	48	3	6	7	9	193
16	217	62	56	17	32	11	33
0	88	66	197	6	0	75	13
213	229	209	213	35	237	176	18
225	17	32	0	25	209	193	16
234	193	13	32	173	62	2	205
1	22	38	0	6	11	62	22
215	124	215	62	21	215	62	143
215	36	16	242	62	63	50	141
92	201						

Table B.

62	40	17	10	8	33	22	88
205	142	249	62	48	17	10	3
33	22	89	205	142	249	62	56
50	141	92	201	66	229	72	67
119	35	16	252	225	213	17	32
0	25	209	65	16	239	201	

Table C.

1	30	4	33	1	80	197	229
229	209	197	33	32	0	68	25
125	254	32	48	4	124	198	7
103	229	237	176	209	193	16	234
65	62	0	18	19	16	252	225
193	36	124	230	7	32	215	201

Table D.

3	7	4	28	63	127	127	63
128	192	224	240	252	255	255	255
0	0	0	0	0	0	192	240
0	0	0	0	0	0	0	0
0	15	7	0	0	0	0	0
247	243	255	255	125	57	63	31
252	255	159	187	255	255	153	254
0	0	192	224	240	240	248	88
0	0	0	0	0	0	0	0
4	7	15	14	28	28	48	99
127	125	125	31	15	3	255	255
152	252	60	252	254	254	254	252
0	0	0	0	0	0	0	0
0	0	7	15	18	13	18	31
0	0	224	240	72	175	72	248
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
56	48	54	96	97	240	115	8
28	12	108	6	134	15	205	16
0	0	0	0	0	0	0	0
0	0	1	3	7	7	15	15
1	251	251	247	255	253	253	240
248	223	223	239	255	191	191	15
0	0	128	192	224	224	240	240

Table E.

D_FILE at row r, column 1 :-

	Address:	low	high
r = 12		129	72
13		161	72
14		193	72
15		225	72
16		1	80
17		33	80
18		65	80
19		97	80
20		129	80

WORLD GAMES

WORLD GAMES
US Gold
£9.99

U.S. Gold go around the world in search of bizarre sporting events

Just when you thought it was safe to pick up a joystick eight new sporting events appear to challenge you in a round the world compilation.

Russia is the first stop and a chance to challenge the champions of the iron game at their own sport. Two different weight lifting events are ready to test your skills, the snatch and the clean and jerk. Up and down joystick movements need to be timed well to complete the lifting movements to ensure that you don't collapse in a heap and get the white lights from the judges.

In competition the key is deciding when to increase the weights (shown on screen as the dumbells get bigger) to knock out your opponents without straining yourself.

After a quick flight to Germany it's time to take part in a 300 year old sport, barrel jumping. The contestants have three attempts to leap the highest number barrels lined up on the ice rink.

Synchronised joystick waggling is the hint for this event to get as much speed as possible so you clear the barrels in time for a good landing. Get it wrong and you'll hurt more than your pride.

Flying all the way to Mexico just to hurl yourself off a cliff may sound over the top but it's because cliff diving is the next event in these global games.

Six ledges test your nerve and your skill as you plummet into the depths of the sea below. If you mistime your leap you could impale yourself on the rocks below. To impress the judges in your three attempts and take the gold you need to dice with danger and dive with style.

The next event is another downhill plummet only this time it's a slalom down the ski slopes of France.

A course fraught with peril and studded with gates to be negotiated really tests your skill. Miss too many gates and the penalty times will put you out of contention but if you hit a gate head on or the side of the course you'll be wiped out!

Next stop is Canada and a round of river log rolling in which you must walk the log so that you're opponent takes the plunge. You must time your joystick moves to keep in step while unbalancing your opponent. Whoever loses his balance first becomes dinner for the sharks. Best of three decides the contest.

Further north in America Ferdinand, Elmer, Bob and Tornado await your arrival. They're bulls that you have to ride in a rodeo. The bulls will try to buck, spin and stop to try and unseat you. You have to counter its moves to stay on as long as you can.

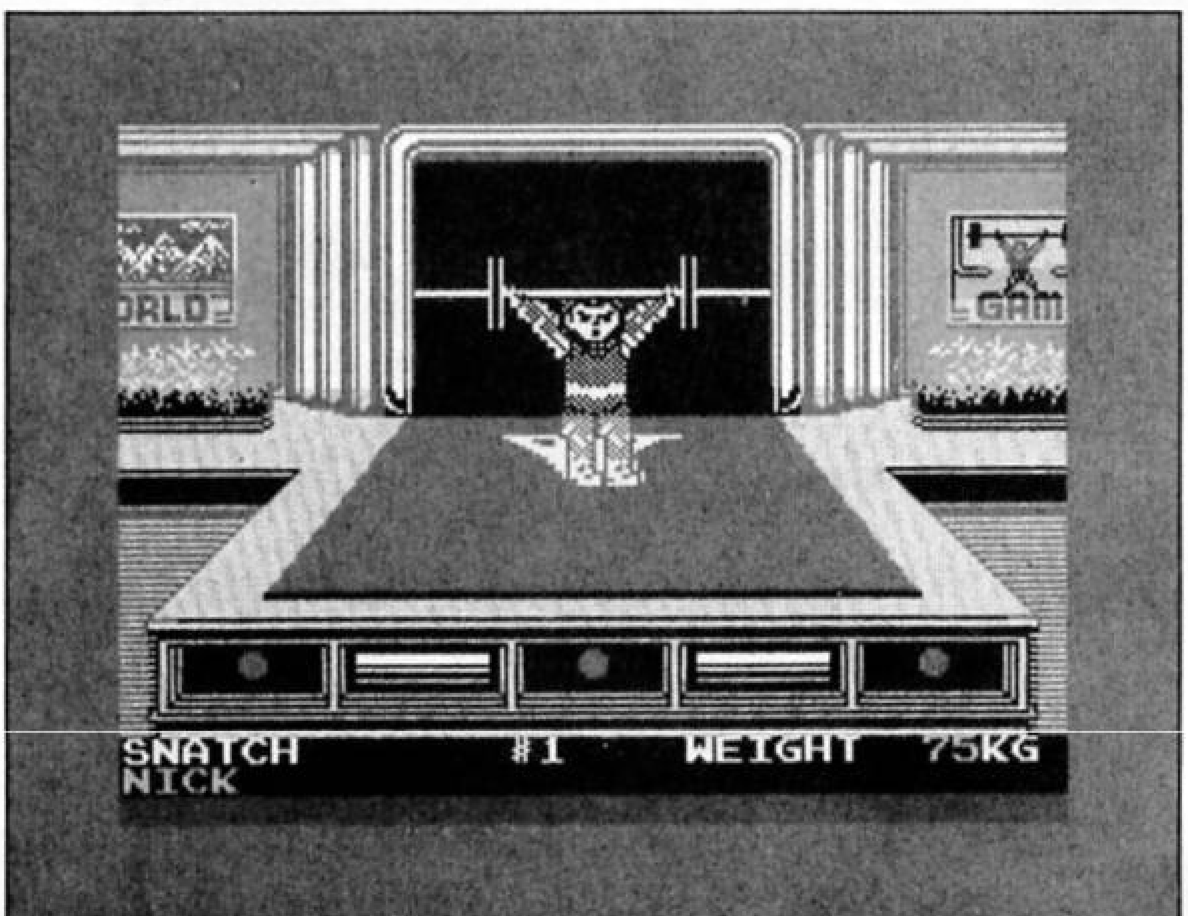
No global games would be complete without a trip to the heather of Scotland and a chance to toss the caber.

Timed left and right joystick movements will start your run (after a quick jig to the bagpipes) and then you must toss the caber so the log flips over one complete turn and doesn't land on your head or your foot! This is probably the hardest event to master and the most painful to watch.

The final stop in our eight event and 24,000 mile trip is the battle of the blubber as two sumo wrestlers battle to shove each other out of the ring.

As the two 400 pound warriors size each other up you have a chance to revise the moves to push, slap and grab your opponent. If executed correctly you could manage a spectacular throw and another gold medal.

World Games is the latest in the series of conversions of Epyx games that featured Summer Games II and Winter Games. This Spectrum version has been coded by Irish programmers, Choice (who were responsible for the superb Hardball) and have added quality to the graphics and animation that was missing from the others. If you're a joystick athlete you can't afford to miss this one.





Specialist REPAIR Guide

Contact Nicola Baty on 01-437 0699

Ring Nicola Baty on 01-437 0699 for details of series discounts

Head

What do Radio Rentals, Foot and Mouth and Batman have in common? Answer: John Ritman, author of the hit game Head Over Heels.

John Ritman doesn't like free newspapers. The first thing I noticed when I arrived on the doorstep of his home in Hendon was a sign saying 'No free newspapers.'

Fortunately he seemed to be a bit fonder of computer mags, which is why I was there to meet the man behind a string of software hits including games such as *3D Combat Zone, Match Day, Batman,* and his latest Monster Hit, *Head Over Heels.*

When I arrived John was experimenting with the sound chip on his ("well dodgy") 128, channelling a tape of The Band through the computer, digitising the music and playing it out through the sound chip via his television speaker. The result was some of the best sound I've ever heard coming out of a Spectrum, but don't expect to hear many symphonies coming out of your machine in the near future because storing that sort of sound gobbles up memory at the rate of 10K a second . . .

Whilst on the subject of sound, John mentioned something that wasn't obvious from looking at the instructions for Head Over Heels, which is that if you've got a 128 machine you can load the game in 128 mode and get extra tunes and sound effects that aren't present in 48K mode.

John was joined by his graphics designer, Bernie Drummond, and we started to chat about the new game which, in its early stages was called Foot and Mouth.

"We were a bit worried that the reaction would be 'Batman mark 2;'" John told me. "But Head Over Heels was a logical progression from Batman. When I finish a game I think I can do a lot better, and the graphics are better because Bernie's learnt a bit since then." Bernied agreed and, pointing at a screen dump of one of the Batman screens, said that he shudders when he looks at it now. Despite the



Programmer

similarities between the two games Head Over Heels is clearly an advance over Batman, with clearer graphics, many more rooms, and the added complexities of controlling two characters rather than just one.

I had assumed that the Batman game was commissioned by Ocean as part of their never-ending quest for licensing deals, but it turns out that Batman was their idea. John had seen Knight Lore and was "mightily impressed", and thought he'd like to produce something like it. According to Bernie the Batman connection was just a 'hook'.

Hooked up

"The thing was, how to avoid ties with Knight Lore, and being labelled an Ultimate clone. So we thought of a hook that had importance of its own." And once they'd decided on Batman, Ocean handled the licensing agreement. It seems that John likes working for Ocean. He says they're very efficient and have much higher sales than most software houses. For most companies a game that sells 12,000 copies is a success, but Batman sold 11,000 in just the two months leading up to Christmas '86, and that was six months after the game was first released. Total sales of Batman are approaching 60,000 while his earlier game, Match Day (written with Chris Clarke) has topped the 100,000 mark.

So, how did John get started on the trail that led to Head Over Heels?

"I worked as a TV field engineer for Radio Rentals. They were doing trials on Atari machines to see if they were worth renting, so I thought it would be worth knowing something about them."

Like so many programmers, John bought a ZX81 when it was first produced in 1982, and his first game, Namtir Raiders came out on the Artic label just four months later.

"I bought a Sinclair games tape and it was dire. I thought 'I've got to be able to do better than that'." Artic gave John a Spectrum and he produced three more games for them before moving to Ocean. A couple of years later came the encounter with Knight Lore.

"I've got a lot of respect for Ultimate. I don't think they're brilliant programmers but they are good games designers. They invented this (3D) format and when I saw it I thought — Wow! I wish I'd done it.

"When I looked at Knight Lore I didn't have a clue how they were doing it but I invented my own way of doing it."

And, if anything, John's method has improved on the original. He claims that his techniques for storing data are more efficient than he's seen in any other programs and it is this which allows him to make Head Over Heels so complex. The game has 300 rooms in it, yet the data to describe all these rooms with their traps and obstacles occupies just 5K. The average room takes up seventeen bytes, while the smallest rooms are a mere seven. It makes the mind boggle to think what he could do using the full 128K memory of the Plus 2, or even the fast access of the planned disc-based Plus 3.

The rest of the game consists of 2K for sound effects, 16-17K of graphics data, and about 20K of code to control the 'mechanics' of the game. John says that he spends a lot of time just working on the basic mechanics of the game, and that the fine details such as the layout of the individual rooms can be added in a relatively short period of time towards the end of the process. When he was writing Batman, John spent three months working on the environment that the game is set within before he even saw anything on the screen.

Once he's got the basics of the game worked out John then tells Bernie roughly what he wants and how much space he's got on screen for the graphics, and Bernie goes off and works on some ideas. After this they work more or less side by side exchanging ideas and making adjustments as they go along.

John likes to put together thinking problems and combine these with a physical problem, so that not only do you have to logically work out the solution to a particular task (such as reaching one of the four crowns on Head Over Heels), you also have the added task of manipulating the objects around you in order to physically put your ideas into action. Sometimes you can see what you want to do but you can't quite work out *how* to do it. It's this two-tiered approach to the gameplay that makes Head Over Heels so horribly addictive.

"You have to envisage how the game is going to play. I do think you have to get the gameplay right in order to make sure you've got something worth playing. Some programmers spend so much time trying to produce a work of technical genius that they lose sight of the gameplay."

Future fixtures

John and Bernie's next project is to help out Match Day co-author Chris Clark with Soccerama, a highly realistic football game designed for up to three players. And this will probably be followed by Match Day 2, a totally new version that will be designed from scratch. Then, looking a bit further into the future, John is thinking about returning to the 3D format with a game that will map itself as you go along, eliminating the need for reams of paper to draw maps on. Being a fan of Dungeons and Dragons he says that he would also like to produce a two-player game in the 3D format, in which the two players can hack it out on screen just as they would in a game of D&D.

I asked John what he does for entertainment apart from playing D&D. Does he play other people's games or avoid computers altogether?

"I program for entertainment." Both John and Bernie get up late, stay up late and work more or less when they feel like it. "I don't work office hours at all. I tend to work 11 am to 4 pm or 11 pm to 4 am. It's a nice life."

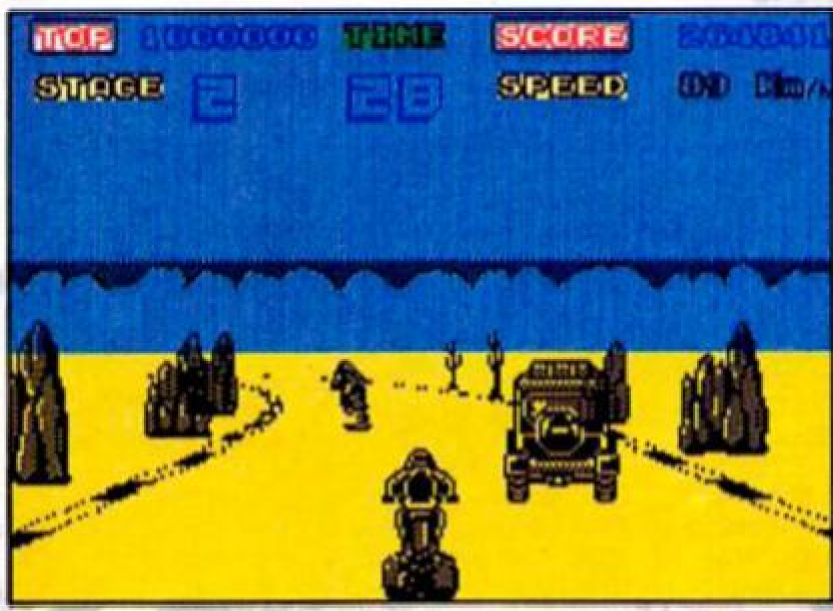
Games by John Ritman

1982: Namtir Raiders (ZX81), Arg (not released).
1983: Cosmic Debris, 3D Combat Zone, Dimension Destructors, Bear Bovver.
1984: Matchday (with Chris Clarke).
1986: Batman (with Bernie Drummond).
1987: Head Over Heels (with Bernie Drummond), Three And In (with Chris Clarke & Bernie Drummond).

ENDURO RACER

Motorcycle mayhem in Activision's dirt track hit.

Enduro Racer
Activision
£9.95





Enduro Racer is quite simply the best motorbike racing game I have ever played and is destined to follow in the tracks of Gauntlet and Commando as a chart-topping coin-op conversion.

The game is a race against other bikes and against the clock as you've just 60 seconds to complete the circuit!

You're moving through the field nicely as you hit the first bend. Then suddenly you notice a pile of rocks across the whole road! Almost without thinking you pull the bike back into a wheelie to leap the barrier and sail over to land safely on the track having just missed a tree and bike busting pile of boulders.

More bends, hills, boulders, trees and jumps lie in your way as the track flows ahead of you in startlingly clear graphics. As our hero navigates a tight bend he puts his leg out to steady the bike and if he comes a cropper the crash is almost painful to watch.

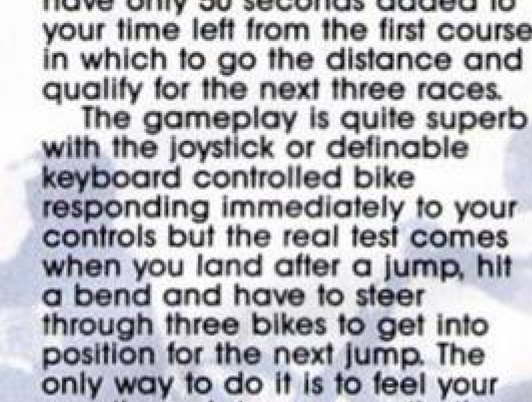
If he runs out of time before he's completed the circuit the game ends with a percentage score showing how much of the course you've navigated, which drives you back for another game especially if you've just missed out by a narrow margin.

Should you succeed then you've no time to celebrate, not even for a lap of honour, as you're literally thrown into the second of the five courses. This time the action's set in a desert with dust clouds streaming from your back wheel as your tear around the course.

In the first race the bikes did little to get in your way but this time you'll have to avoid land rovers trying to mow you down as well as more jumps, fallen trees in the middle of the road and massive boulders that almost block the road entirely!

To add to your problems you have only 50 seconds added to your time left from the first course in which to go the distance and qualify for the next three races.

The gameplay is quite superb with the joystick or definable keyboard controlled bike responding immediately to your controls but the real test comes when you land after a jump, hit a bend and have to steer through three bikes to get into position for the next jump. The only way to do it is to feel your way through in a game that's so realistic you'll almost need a crash helmet!



THE DISCOVERY COLUMN



More advice and readers' tips, presented by John Wase.

For some time now I have had random access file programs coming in. Two are particularly worthy of mention this month. Firstly, a program from Andrew Bailey of Bradford which sets up a name and address system (Oh no! I hear you cry) but with a difference: each segment is on a separate random access file. It's a very long program, but I hope to include it in a future issue. We do have problems with long programs; please do try and keep them reasonably short if possible. One solution is to include a program segment, and this is what I propose to do with the second item, a listing from Brian Coutts of East Kilbride. At last we have some novel programs coming up, for Brian searches random access files with this binary search routine (figure 1).

Don't try and RUN this program; it won't. It is merely a segment of a 17K program and it demonstrates binary search principles rather nicely. Let's start with the 17K program: it's a Pools Forecasting Program which accesses a number of random access files. These include a history file (1237 records, each of 15 characters (14 + "ENTER"). 1-3 = 1st team number; 4-6 = 2nd team number; 7-10, previous results at 1st team ground; 11-14, previous results at second team's ground). In addition to the history file, there is a team file (also random access) with the team names and numbers, and when the fixtures array is set up, this is also saved as a random access file. The program will also shortly be expanded to include league table information to assist the forecast.

The listing (figure 1) is, I repeat, merely a search routine. It is, however, very well documented with REM statements, and there should be no difficulty in following its operations; it searches for the history or past performance of each of the two teams in the current fixture. This enables the main program to predict a win, a lose or a draw. Brian adds that this program was started only this year, and to date he has one small win. So he gives no guarantees as to instant riches...

Tasword

Mr. Goodman of Stoke-on-Trent sent me an interesting letter supplementing our recent article on "the better letter". He sent a listing to alter the on-screen character shapes for the Opus version of Tasword+2. Enter **CLEAR 25299**, and load the Tasword+2 code ("TASCODE"CODE 23500). Put your disc carefully away. Use the four line loader in figure 2 to POKE in the appropriate values; save the modified code on a new disc with **SAVE * 1;"TASCODE" 23500, 40236**. Mr. Goodman also writes that he has suffered my +2 problems (see my recent letter); he too complains of lack of cursor movement when the +2 is cold, although it's fine in BASIC, and fine when it warms up. I wonder if this is a problem specific to the Opus power unit or the Opus Tasword version? Anyone out there got a solution yet? In addition, he had

Fig. 1 Binary search routine.

```
   1 REM ****** BINARY  ******
   2 REM ***SEARCH  ROUTINE***
   3 REM ***BRIAN J. COUTTS***
   4 REM
2000 REM find & fill in history
2010 LET v=1: LET u=0: LET fixtu
res=u
2020 CLS : PRINT "Filling in fix
tures history.": PRINT "Please w
ait for the BEEP!"
2030 OPEN #5;"m";1;"historyRND"R
ND15: REM random access file-123
7 records,15 characters each.
2040 FOR n=v TO f: LET e=u: LET
e$="": REM sets up variables for
 search(f is the no.of fixtures)
.
2050 IF VAL f$(n,v TO 3)>VAL f$(
n,34 TO 36) THEN LET e=v: LET e$
=f$(n,34 TO 36)+f$(n,v TO 3): GO
 TO 2070: REM f$ is fixtures arr
ay.Characters 1-3 and 34-36 are
home and away team nos.Lines 205
0/2060 puts them into numerical
order.
2060 LET e$=f$(n,v TO 3)+f$(n,34
 TO 36)
2070 REM search for e$(v TO 3)
2080 LET a=v: LET b=1237: LET se
arch=u: REM sets initial values
for each search.It is a binary s
earch which tries the first,last
 and middle,then resets the vari
ables depending on what it finds
.
2090POINT #5;a: INPUT #5;h$: IF
h$(v TO 3)=e$(v TO 3) THEN LET s
earch=a: GO SUB 2210: GO TO 2160
: REM if 1st team found then go
to routine to find 2nd team & th
en start again for the next.
2100POINT #5;b: INPUT #5;h$: IF
h$(v TO 3)=e$(v TO 3) THEN LET s
earch=b: GO SUB 2210: GO TO 2160
2110 LET search=INT (.5+(b+a)/2)
2120POINT #5;search: INPUT #5;h$
: IF h$(v TO 3)=e$(v TO 3) THEN
GO SUB 2210: GO TO 2160
2130 IF VAL h$(v TO 3)>VAL e$(v
TO 3) THEN LET b=search
2140 IF VAL h$(v TO 3)<VAL e$(v
TO 3) THEN LET a=search
2150 GO TO 2090
2160 NEXT n: CLOSE #5
2170 PRINT #v;"History entered."
: LET fixtures=v
2180 BEEP 5,1: RETURN
2190 REM search for e$(4 TO 6).A
fter having found the correct ar
ea of the history file the searc
h proceeds 1 record at a time un
til it finds the correct record.
2200POINT #5;search: INPUT #5;h$
2210 IF VAL h$(4 TO 6)=VAL e$(4
TO 6) THEN GO TO 2240
2220 IF VAL h$(4 TO 6)<VAL e$(4
TO 6) THEN LET search=search+1:
GO TO 2200
2230 LET search=search-1: GO TO
2200
2240 REM enter history to fixtur
e.Having found the correct recor
d the history is entered into th
e fixtures array and returns for
 the next fixture.
2250 IF e=v THEN LET f$(n,7 TO 1
0)=h$(11 TO 14): LET f$(n,40 TO
43)=h$(7 TO 10)
2260 IF e=u THEN LET f$(n,7 TO 1
0)=h$(7 TO 10): LET f$(n,40 TO 4
3)=h$(11 TO 14)
2270 RETURN
```

Fig. 2 New Tasword characters.

```
10 Input "Address ";ad: PRINT ad,
20 Input "Value ";va: PRINT va
30 POKE ad,va
40 GOTO 10
```

Address	Value	Address	Value	Address	Value
55050	7	55106	7	55162	7
55052	7	55107	5	55166	7
55058	7	55112	2	55170	7
55059	5	55113	0	55173	7
55062	7	55114	6	55178	7
55066	7	55115	2	55181	7
55070	7	55120	1	55186	7
55074	7	55121	0	55187	4
55075	5	55122	1	55194	7
55078	7	55127	3	55196	7
55082	7	55137	6	55198	7
55084	7	55138	2	55206	3
55086	7	55139	2	55227	5
55089	6	55140	2	55230	5
55098	7	55141	2	55245	7
55101	7	55142	7	55247	7
55103	7	55149	5		

problems with the T2T3 program which did not always respond to his selection when saving (Press D or T); inserting the line **165 IF INKEY$<>"" THEN GO TO 165** solved the problem.

Moving story

Whilst I'm talking about problems, I had one the other day. I use Jose Pedro's 48K program "Opuscat2" as a disc organiser. Amongst many other features it moves files from one disc to another. When I used it to transfer Tasword+2, a 128K program, I lost a piece of code. I always assumed that 'MOVE a file" put a piece of program into some buffer and then outputted it onto another disc, so it didn't matter if it was in 48K or 128K mode. Does it? Does anyone really know how 'MOVE' works? Meanwhile, I advise caution: make sure that everything you wanted is moved and checked before anything is erased.

Hidden messages

Now a nice little quickie from Rene Uittenbogaard, Nijmegen, Netherlands; (figure 3). Tap it in, run it and you will get the undocumented error message "N Wrong Disc", which, Rene says, appears when one tries to move RAM disc to a normal disc with **"MOVE "d";5 TO "d";1"**. I would add that it also appears

```
  10 RESTORE : FOR f=USR "a" TO
USR "a"+3: READ a: POKE f,a: NEX
T f: DATA 205,8,23,207
  15 POKE USR "a"+4,54: RANDOMIZ
E USR USR "a"
```

Fig. 3 Program to display error messages.

when you try and copy one disc to another when the discs are of different sizes. Rene also mentions that changing the 54 in line 15 to 71 gives "©Dave Corney". He or course wrote the ROM. Changing it to 72 gives a surprise error message. In fact you can open a stream to a stream with **OPEN#3;#3**. But if you then try to **PRINT#3** — up comes your surprise message.

Shinwa dump

Screen dumps are coming in thick and fast. Ejgil Hansen of Karlslunde, Denmark, wrote in with this neat machine code program to do a screen dump on a Shinwa CP80 from the Opus port.

The machine code is given in decimal format in figure 4. CLEAR 51999 and POKE each number into the appropriate address, starting at 52000; the left hand column lists every 10th address and to the right are the ten numbers to be POKEd into the ten addresses. Alternatively, you can write your own loader program to do it. The checksum is 68948. Type in the short bit of BASIC (figure 5) to RUN it and all should be well. Ejgil says it

Fig. 4 Machine code for Shinwa dump.

CODE 'MCOPY'52000,674

52000	17	0	3	33	248	207	1	254	231	229
52010	62	0	203	126	40	2	203	255	35	203
52020	126	40	2	203	247	35	203	126	40	2
52030	203	239	35	203	126	40	2	203	231	35
52040	203	126	40	2	203	223	35	203	126	40
52050	2	203	215	35	203	126	40	2	203	207
52060	35	203	126	40	2	203	199	2	3	225
52070	229	62	0	203	118	40	2	203	255	35
52080	203	118	40	2	203	247	35	203	118	40
52090	2	203	239	35	203	118	40	2	203	231
52100	35	203	118	40	2	203	223	35	203	118
52110	40	2	203	215	35	203	118	40	2	203
52120	207	35	203	118	40	2	203	199	2	3
52130	225	229	62	0	203	110	40	2	203	255
52140	35	203	110	40	2	203	247	35	203	110
52150	40	2	203	239	35	203	110	40	2	203
52160	231	35	203	110	40	2	203	223	35	203
52170	110	40	2	203	215	35	203	110	40	2
52180	203	207	35	203	110	40	2	203	199	2
52190	3	225	229	62	0	203	102	40	2	203
52200	255	35	203	102	40	2	203	247	35	203
52210	102	40	2	203	239	35	203	102	40	2
52220	203	231	35	203	102	40	2	203	223	35
52230	203	102	40	2	203	215	35	203	102	40
52240	2	203	207	35	203	102	40	2	203	199
52250	2	3	225	229	62	0	203	94	40	2
52260	203	255	35	203	94	40	2	203	247	35
52270	203	94	40	2	203	239	35	203	94	40
52280	2	203	231	35	203	94	40	2	203	223
52290	35	203	94	40	2	203	215	35	203	94
52300	40	2	203	207	35	203	94	40	2	203
52310	199	2	3	225	229	62	0	203	86	40
52320	2	203	255	35	203	86	40	2	203	247
52330	35	203	86	40	2	203	239	35	203	86
52340	40	2	203	231	35	203	86	40	2	203
52350	223	35	203	86	40	2	203	215	35	203
52360	86	40	2	203	207	35	203	86	40	2
52370	203	199	2	3	225	229	62	0	203	78
52380	40	2	203	255	35	203	78	40	2	203
52390	247	35	203	78	40	2	203	239	35	203
52400	78	40	2	203	231	35	203	78	40	2
52410	203	223	35	203	78	40	2	203	215	35
52420	203	78	40	2	203	207	35	203	78	40
52430	2	203	199	2	3	225	229	62	0	203
52440	70	40	2	203	255	35	203	70	40	2
52450	203	247	35	203	70	40	2	203	239	35
52460	203	70	40	2	203	231	35	203	70	40
52470	2	203	223	35	203	70	40	2	203	215
52480	35	203	70	40	2	203	207	35	203	70
52490	40	2	203	199	2	3	225	27	122	179
52500	200	35	35	35	35	35	35	35	35	195
52510	41	203	42	127	69	74	71	73	76	42
52520	22	0	1	255	63	33	248	207	3	197
52530	10	119	4	35	10	119	4	35	10	119
52540	4	35	10	119	4	35	10	119	4	35
52550	10	119	4	35	10	119	4	35	10	119
52560	4	35	193	20	122	178	40	3	195	40
52570	205	22	0	1	255	71	33	248	215	3
52580	197	10	119	4	35	10	119	4	35	10
52590	119	4	35	10	119	4	35	10	119	4
52600	35	10	119	4	35	10	119	4	35	10
52610	119	4	35	193	20	122	178	40	3	195
52620	99	205	22	0	1	255	79	33	248	223
52630	3	197	10	119	4	35	10	119	4	35
52640	10	119	4	35	10	119	4	35	10	119
52650	4	35	10	119	4	35	10	119	4	35
52660	10	119	4	35	193	20	122	178	40	3
52670	195	150	205	201						

Fig. 5 Program to run Shinwa dump.

```
  1 REM COPY
  2
  3 CLEAR 51999
  4 LOAD *"m";1;"MCOPY"CODE 52000,674
  5
  6 INPUT "SCREEN NAME ";Q$
  7 LOAD *"m";1;Q$SCREEN$
  8 RANDOMIZE USR 52520
  9 RANDOMIZE USR 52000
 10 OPEN #5;"b"
 20 INPUT "TABULATION ";tab
 30 FOR y=0 TO 21
 40 PRINT #5;CHR$ 27;"@";
 50 PRINT #5;CHR$ 27;"D";CHR$ tab;CHR$ 0;CHR$ 9;
 60 PRINT #5;CHR$ 27;"U";CHR$ 1;
 70 PRINT #5;CHR$ 27;"K";CHR$ 0;CHR$ 1;
 80 FOR x=0 TO 255
 90 PRINT #5;CHR$ (PEEK (59390+y*256+x));
100 NEXT x
110 PRINT #5;CHR$ 27;"J";CHR$ 24;
120 NEXT y
130 PRINT #5;CHR$ 27;"@";
140 CLOSE #5
```



Fig. 6 Sample Shinwa screen dump.

copies in about 2 minutes, but I couldn't test it: my wife has a Shinwa, but it saw me coming and went on strike. Ejgil also asks if anyone knows how to repair a sick disc with only one answer — I/O error. My own feeling is that since it's having trouble reading even the catalogue file, then either this has been totally corrupted or the sector markers have been lost. If it is merely the catalogue that has been corrupted, then you might win with the "read block" facility in Bradway Software's Disc Manager which allows you to read in any block from the disc into a buffer area in the Spectrum, modify it, and save it on another disc.

Epson dump

Next, Patrick Mackinlay of Induno Olona, Italy, also sends in a machine code dump — this time for an Epson (figure 7). It is stored in the printer buffer, so we don't need to CLEAR anything, and it must be a 48K Spectrum or a 128 in 48K mode. Type it in using an assembler. To get the listing to work, Patrick says, you need to **OPEN #3;"b"** and then call it with **RANDOMISE USR 23339.**

CAT's back

Finally, a program by M. D. Wood of Larkfield, Kent. This arrived a couple of months ago, but somewhere along the line, all the documentation disappeared — I could find only a disc, Mr. Wood's name and address, and a very brief listing. So I put it on one side until I had more time; a fatal move. When I at last looked at it, I thought "Oh no — just another CAT program". Then I looked again, and found that it used Discovery's RAMdisc facility and got intrigued. It printed out a catalogue with additional information; not, perhaps, as nicely as "fullcat" or "ecat", which have already been published, but it uses the RAMdisc rather neatly. Just type in the listing in figure 8 and see. I know it's not very exotic, but I can't recall any other program which has used this facility, and it is pretty useful after all. There

Fig. 7 Epson dump.

```
                 10 *S
5B00             20         ORG  23296


        Copy to Epson printer routine

                 30 *H             Copy to Epson printer routine
                 40
                 50 ;Copy line routine
5B00 012008      60 LCSrt  LD   BC,#0820 ;8 (pixels) * 32 (codes) per line
5B03 1E40        70        LD   E,%1000000
5B05 1600        80 SILp   LD   D,0       ;D will = byte for printer
5B07 E5          90        PUSH HL
5B08 7E         100 ILoop  LD   A,(HL)    ;A=byte from display file
5B09 A3         110        AND  E         ;A =special pixel
5B0A FE00       120        CP   0         ;jump if pixel = 0
5B0C 2802       130        JR   Z,Reset
5B0E CBC2       140        SET  0,D        ;set pixel
5B10 CB02       150 Reset  RLC  D         ;nextpixel
5B12 24         160        INC  H
5B13 10F3       170        DJNZ ILoop     ;5 pixels per byte
5B15 CB0A       180        RRC  D         ;D=byte for printer
5B17 7A         190        LD   A,D       ;lprint D
5B18 F5         200        PUSH AF
5B19 D7         210        RST  #10
5B1A F1         220        POP  AF
5B1B D7         230        RST  #10
5B1C E1         240        POP  HL        ;reset      HL
5B1D CB0B       250        RRC  E         ;next pixel from display file
5B1F 3E80       260        LD   A,%10000000  ;have all 8 pixels been taken?
5B21 BB         270        CP   E         ;if not continue
5B22 0608       280        LD   B,8
5B24 20DF       290        JR   NZ,SILp
5B26 23         300        INC  HL        ;nextbyte in display file
5B27 0D         310        DEC  C         ;have all 32 bytes been taken?
5B28 20DB       320        JR   NZ,SILp  ;if not continue
5B2A C9         330        RET



                340 *H
                350 ;Main routine:
5B2B            360        ENT  $
5B2B 3E03       370 MStrt  LD   A,3       ;set RST 10 to stream 3(printer)
5B2D CD0116     380        CALL ChanOp
5B30 21E03F     390        LD   HL,#4000-#20 ;beginning of display-#20
5B33 01735B     400        LD   BC,HLvals
5B36 C5         410        PUSH BC
5B37 E5         420        PUSH HL
5B38 3E18       430        LD   A,24
```

▶ ▶ ▶

```
5B3A 32685B     440 MLoop  LD   (lines),A
5B3D 21695B     450        LD   HL,CommdC    ;HL=Address of control codes
5B40 060A       460        LD   B,10      ;7 control codes
5B42 7E         470 CmndCL LD   A,(HL)    ;send control codes
5B43 D7         480        RST  #10
5B44 23         490        INC  HL
5B45 10FB       500        DJNZ CmndCL
5B47 E1         510        POP  HL
5B48 C1         520        POP  BC
5B49 112000     530        LD   DE,#20    ;next        line
5B4C 19         540        ADD  HL,DE
5B4D 0A         550        LD   A,(BC)
5B4E BC         560        CP   H
5B4F 2004       570        JR   NZ,Simple
5B51 03         580        INC  BC
5B52 0A         590        LD   A,(BC)
5B53 67         600        LD   H,A
5B54 03         610        INC  BC
5B55 C5         620 Simple PUSH BC
5B56 E5         630        PUSH HL
5B57 CD005B     640        CALL LCSrt     ;lprint line
5B5A 3A685B     650        LD   A,(lines);
5B5D 3D         660        DEC  A
5B5E 20DA       670        JR   NZ,MLoop
5B60 3E02       680        LD   A,2
5B62 CD0116     690        CALL ChanOp
5B65 E1         700        POP  HL
5B66 D1         710        POP  DE
5B67 C9         720        RET  ;return if all screen copied
1601            730 ChanOp EQU  #1601


                740 *E
                750 ;system variables
5B68 18         760 lines  DEFB 24        ;lines to be copied
5B69 00000A     770 ComndC DEFB 0,0,10    ;line feed
5B6C 1B4108     780        DEFB 27,65,8  ;ESC A,8
5B6F 1B4C0002   790        DEFB 27,76,0,2;ESC C,2*256
5B73 41484950   800 HLvals DEFB #41,#48,#49,#50

Pass 2 errors: 00

Table used:   161  from   323
Executes: 23339
```

must be other RAMdisc applications, so let's see some programs using this principle in the next few months: do sent them in if you have written something useful, and *reasonably short* and snappy.

See you next month then, and keep the programs coming.

Fig. 8 Cat program by M. D. Wood.

```
   1 REM *** TRUE CAT ***
         *** M.D.WOOD ***
   2 CLEAR 32767: FORMAT 5;"RAMD
ISC": POKE 23658,8: LET P=2: CLE
AR #: OPEN #3;"T";ORND80: INPUT
"SCREEN OR PRINTER :P/S "; LINE
P$
   3 IF P$="P" THEN LET P=3
   4 IF P$="S" THEN LET P=2
   5 DEF FN a()=FN b()+256*FN b(
): DEF FN b()=CODE INKEY$#4
  10 OPEN #5;5;CHR$ ORND1,-1: IN
PUT "DRIVE : ";D: CAT #5;D:POINT
 #5;13
  20 INPUT #5; LINE f$: IF f$=""
 THEN STOP
  30 CLEAR #4: OPEN #4;d;f$: LET
 t=FN b(): LET l=FN a(): PRINT #
P;f$;" ";
  40 IF t=0 THEN LET n=FN a(): P
RINT #P;"PROG ";L;",";FN A();" L
INE "+STR$ N AND N<4^7;
  50 IF T=1 OR T=2 THEN PRINT #P
;"DATA ";CHR$ (FN B()*0+FN B()-6
4*T+32);"$" AND T=2;: FOR J=1 TO
 FN A()*0+FN B(): PRINT #P;"(,"(
1+(J>1));FN A();: NEXT J: PRINT
#P;")";
  60 IF T=3 THEN PRINT #P;"CODE
";FN A();",";L;
  70 PRINT #P: GO TO 20
```

ZX COMPUTING

FULL CREDIT FACILITIES FOR ACCESS AND BARCLAYCARD.

Lineage: 48p per word. (VAT inclusive)

Semi display: £9.50 per single column centimetre + VAT. Ring for information on series bookings/ discounts (minimum 2.5cm).

All advertisements in this section must be prepaid. Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01 - 437 0699

Send your requirements to:
NICOLA BATY
ASP Ltd., 1 Golden Square, London W1

UTILITIES

BRADWAY SOFTWARE (ZX)

LETTA-HEAD PLUS The most versatile Spectrum utility program for designing and printing business and personal stationery, receipts, letterheads, labels, posters, etc. Create the design on screen using inbuilt graphics or load any SCREENS from an arts package, add text from the library of 25 different fonts, select the required output format and print all the copies you need. **Price only £9.00**

DUMPY 3 Fast becoming the de facto standard screendump program, producing all the screen dumps you will ever need for your Spectrum. No need to understand assembler: **DUMPY's code GENERATOR** creates the screendump, relocates it, and saves it ready for use in your programs. Over 1000 different sizes, shape and density screendumps, anywhere in RAM! Price less than 1p per dump!! **£8.50**

LIN-O-TYPE Add style to your written word. Print out any wordprocessor file on tape, microdrive or disc in high denstiy NLQ in a choice of 25 fonts, or turn your Spectrum into a full WYSIWYG electronic typewriter, superb for addressing envelopes, filling in forms or writing short notes. **Price £8.50.**

ASTRUM + NEW!! The very best Z80 assembler and monitor package for the Spectrum: Exceedingly versatile, powerful and easy to use: Full wordprocessor style editor; Unlimited source code; Assembly using microdrives or the Opus Discovery; Command driven monitor with multiple breakpoints, single stepping, slow run etc. **Price £13.50** (state microdrive or Opus version).

DISC MANAGER NEWEST!! A brand new utility for users of the Discovery drives: Full disc sector editor and file management program: Erase, rename, copy groups of files; Examine, modify and understand disc directory information; Examine, modify and recover damaged data. **Price £10.50** including 3½" disc.

Bradway Software programs drive most (currently 16 different) interfaces: **Letta-Head, Dumpy & Lin-O-Type** require an Epson compatible printer and are compatible with Microdrive, Wafadrive & Disc. Prices are for cassettes, include full instruction and P&P within Europe; please add £2.00 per program worldwide airmail postage, £1.50 for microdrive cartridge or disc.

Send for further details!

33 Conalan Avenue, Sheffield S17 4PG

PRINTER SERVICES

FABRIC RIBBON RE-INKING
Introductory Offer
£1.50
Send ribbons and prepayment to:
ALADDINK,
(Dept ZXC), 4 Harker Crescent, Eyemouth, Berwickshire TD14 5AP **Tel: 08907 50965**
Colour re-inking now available
See 'NEWS' ZX April 1987

USE THIS SPACE FOR YOUR SUMMER PLANS
01-437 0699

FOR SALE

SPECTRUM 128K SYSTEM: Interface-I Microdrive, Multiface-I, Kempston Mouse, Kempston-E, Joystick Interfaces, Games, Serious Software. Worth £600. Quick sell £280 incl. free noisy Opus 128K/48K Disc Drive. Write: S.C.Lim, Wilson House, 38-76 Sussex Gardens, London W2 1UF.

WANTED

WANTED: YOUR non working Commodore Computers and equipment for cash. Ring D.M.R. Electronics, Bradford, (0274) 882892.

SERVICES

YOU CAN TURN your microcomputer into a money making machine. Free details from: Oasis Books (ZX1), 7 Cortis Avenue, Worthing, Sussex. BN14 7BQ.

DEADLINES

15TH JUNE
AUGUST ISSUE

SOFTWARE

X RATED
ADULTS ONLY GAME

FANTASY

Spectrum 48/128 ..... £5.50 inc. p&p.
Commodore 64/128 ... £6.50 inc. p&p.
Available to persons over 18 only.
Please state age when ordering.

R 'n' H MICROTEC
32 Hazell Way, Stoke Poges, Bucks SL2 4DD.

HUMPHREY SOFTWARE
TWO PROGRAMS

SUPER HANDICAP; The connoisseur racing program that will pay for itself. Full instructions and recommendations.
ONLY £9.95.

SEEK & FIND; Try to sink your opponents ships before he gets yours, but watch out for the mines! One or two players, many features.
ONLY £3.99

SPECIAL OFFER
BOTH PROGRAMS AT ONLY £11.95 INCL.

HUMPHREY SOFTWARE
8 Richard Andersen Court, Monson Road, London SE14 5EJ.

FORTRAN 77S COMPILER for 48K Spectrum for just £15. Send cheque to Mira Software, 24 Home Close, Kibworth, Leics. LE8 0JT. Standard Pascal Compiler coming soon. Write for further information.

TRICKS OF THE TRADE: Booklet revealing the experts' secrets. Packed with hints, tips & routines to improve your programs. Make programs unhackable. + Much more! Only **£3**

RETURN TO CIVILISATION: Quill written adventure. Escape from the desert. **£2.50**

ANAGRAMS: 3200 7-letter words: **£4**

8-LETTER ANAGRAMS: 5600 8-letter words. Two cassettes: **£5**

ANAGRAM programs are user-friendly and fun to play. Improve your word-power. Ideal for Scrabble players and word enthusiasts.

BARGAIN! ANAGRAMS & 8-LETTER ANAGRAMS: **£8**

(48K Spectrum) From: **I. Rogers, 32 Broadleas Crescent, Devizes, Wiltshire SN10 5DH**

STUDYING FOR THEORY of music exams? Try my practice programs! (Grades 2-5). Cassette £5, Microdrive (2 cartridges) £8. Ted Kirk, 33 Humber Crescent, St. Helens, Merseyside WA9 4HD. Details free.

CHEZRON SOFTWARE
(Dept. 303) 605 Loughborough Road Birstall LEICESTER LE4 4NJ

BONABANNER — Programmers! Unleash the power of string graphics on your Spectrum. Create title pages, text adventure graphics, program headers etc. that print instantly from Basic or machine code.

At just £3.50 why hesitate! — if our demo program doesn't convince that **BONABANNER** is useful to you, JUST RETURN THE WHOLE PACKAGE FOR A FULL REFUND!!

01-437 0699
= RESULTS!

QL/SPECTRUM UTILITIES! Wide range of business/practical programs available. SAE/IRC for details (state Micro). S.D. Microsystems (ZX) PO Box 24, Hitchin, Herts.

TERMS & CONDITIONS

CLASSIFIED ADVERTISING TERMS & CONDITIONS

Our terms for new advertisers (semi-display and lineage) are strictly pro-forma payments until satisfactory reference can be taken up (excluding recognised advertising agencies). Cheques and P.O.'s should be crossed and made payable to ARGUS SPECIALIST PUBLICATIONS LTD. and sent together with the advertisement to:

"The Classified Dept.,
No. 1 Golden Square,
London W1R 3AB.

There are no reimbursements for cancellations. Advertisements arriving too late for a particular issue will be inserted in the following issue unless accompanied by instructions to the contrary. It is the responsibility of the advertiser to ensure that the first insertion of every series is published correctly, and corrections must be notified in time for the second insertion, otherwise the publishers will not accept any liability or offer any reduction in charges.

All advertising sales are subject to Government Regulations concerning VAT. Advertisers are responsible for complying with the various legal requirements in force eg: The Trade Description Act, sexdiscrimination act & the business advertisements (disclosure) order 1977.

Full Terms & Conditions of Advertising available on request.

WARNING NOTICE

Advertisements placed in this magazine are to be in strict compliance with our standard conditions (copies of which conditions are available on request) and on the clear understanding that the advertiser warrants that his advertisement(s) does not infringe any copyright or condition of sale of any interested party in the advertised product.

Further, the advertiser indemnifies the proprietors of this magazine in respect of costs, damages, or any other claims brought against them as a result of legal action arising from the publication of the advertisement.

Any breach of these terms or the said conditions may result in prosecution of the advertiser by the proprietors.

It's not just beer that comes in six-packs.

THE HIT PACK

6 Pak
Hit Pack
£9.95

Depending on your point of view, this is either a six pack compilation with one free game or a seven game compilation. The seventh game in question is Duet and presumably Hit Pak (Elite under another name) are calling it a free game by virtue of the fact that it's not been previously released. Either way this is still a good value for money collection, aimed primarily at the arcade end of the market.

The first game on side one of the tape is the much hyped *Scooby Doo*. When this was announced there was a lot of talk about how marvellous it was going to be, then it seemed to be scrapped because the programmers couldn't fit the game into just 48K, then when it did appear it vanished almost without trace. That was inevitable I suppose, because after all that fuss the end result was an amusing and mildly addictive game with some nice graphics, but it wasn't anything particularly special.

Leaving Scooby to the task of rescuing Shaggy and Co. from the clutches of the mad scientist, you can move on to *1942*. Elite's conversion of the coin-op game in which you have to fight your way through waves of enemy fighters to reach safety on an aircraft carrier in the Pacific. The game did look a bit old fashioned by the time it reached the Spectrum, but it's a nice simple shoot 'em up, the sort that's always good for whiling away a few spare minutes.

Next on the tape is *Duet*, which turns out to be a *Gauntlet* variant, but dressed up in modern soldier gear rather than Gauntlet's wizards 'n warriors trappings. Like Gauntlet it offers a two player option which is much more fun than playing on your own. It's quite good fun, but the Gauntlet similarities are so obvious that I can understand why there was no point in releasing it after the initial flurry of Gauntlet clones had been and gone.

Game number four is *Jet Set Willy II*, the expanded version of the original *Jet Set Willy* game. It's possible that there's a whole generation of relatively new Spectrum owners who have never seen or played JSW or its famed predecessor, *Manic Miner*. However JSW still holds its place as the ultimate platform game (whatever happened to Matthew Smith?).



Split personalities

Side two gets off to a good start with Palace software's *Sacred Armour of Antiriad*. I'm surprised to see this on a compilation so soon, and I wonder how well it sold on its first release?

Antiriad is an animated arcade adventure in which you have to guide the warrior Tal through a ruined city and into the heart of a volcano to destroy a nuclear generator. Along the way you have to collect the components that activate the suit of armour which will protect you from the radiation spillage and the robotic guards that you'll encounter. The graphics are excellent and the gameplay is well thought out, and I'd probably pick this as being the best game on the tape.

Melbourne House's *Fighting Warrior* followed the success of *Exploding Fist*, and while it wasn't quite as addictive as Fist, it's still as good as most of the other martial arts games that tried to copy it, and which are still being churned out even now.

Rounding the tape off is *Split Personalities*, Domark's sliding squares puzzler. Despite being based on a very simple and very old type of puzzle, Domark's computer version managed to be surprisingly enjoyable to play and stands as one of their better efforts.

Apart from Antiriad none of these games are really outstanding, but unlike a lot of compilations which include just a couple of good games and are padded out with a few complete flops, all the games on this six (or seven) pack are good solid arcade entertainment and the combination of all of them together makes this one of the best value arcade compilations I've seen for a while.




THE TUBE
ENTER THE TUBE IF YOU DARE!
SURVIVE THE TRANSFER ZONE, STEER THROUGH THE INNER TUBE DEFENCE MECHANISM TO REACH THE CAPTURE AREA.
THE TUBE IS AN AMAZING FAST ACTION SHOOT EM UP, WITH SUPERB GRAPHICS AND SOUND.
ONLY THOSE WITH NERVES OF STEEL SHOULD APPLY.
Argus Press Software Group
Victory House, Leicester Place, London, WC2H 7NB Tel: 01-439 0666
CBM 64 and Spectrum:
£8.95
QUICKSILVA


PIRATE.GAMBLE.SMUGGLE.LOOT–TO BECOME..
TAI-PAN
COMMODORE £8·95
AMSTRAD £8·95
SPECTRUM £7·95
ATARI ST £14·95
IBM/AMSTRAD PC+COMPATIBLES–£19·95
Now from the author of top seller SHOGUN, James Clavell, comes TAI PAN and a terrific computer game simulation packed with action and stunning graphics, TAI PAN is the exciting story of a man and an island. Become Dirk Struan – a pirate, a smuggler, a manipulator of men achieving riches beyond imagination. Enter a world of blood, sin treachery, conspiracy and murder – a game of Grand entertainment!
Screen shots taken from Atari version.
ocean
Copyright © 1986 James Clavell. Tai Pan is a registered Trademark. © 1986 de Laurentiis Entertainment Inc.
6 Central Street · Manchester · M2 5NS Tel: 061 832 6633 · Telex: 669977